

“Advancing Traffic Efficiency and Safety
through Software Technology phase 2 (ATESST2)”

EAST-ADL2 Requirements

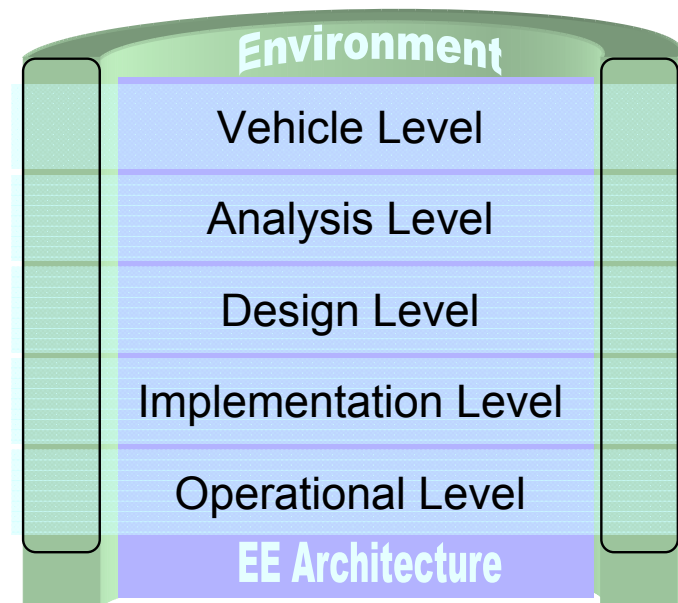
ATESST2 Concept presentation 2009 Q4



General

An EAST-ADL2 model represent the system using several parallel models, each on a different abstraction level

The models on each abstraction level are complete, and represents the EE system “from input to output”



EAST-ADL Structure

Vehicle Level

- Feature content in a VehicleFeatureModel

Analysis Level

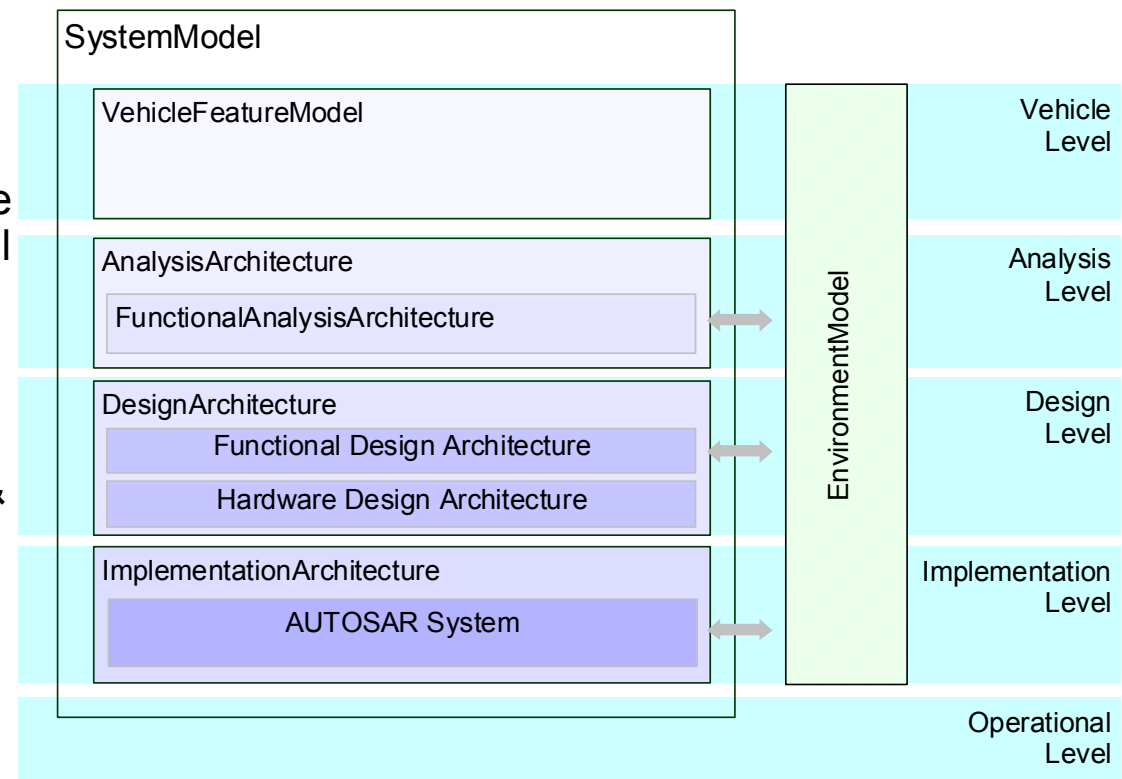
- Functional Analysis Architecture capturing the abstract functional behavior

Design level

- Hardware entities/topology
- Concrete Functional structure & behavior
- Function-to-ECU allocation

Implementation Level

- AUTOSAR constructs



Requirement Representation

To provide language means to

- specify required properties of the system (at varying degrees of abstraction)
- trace requirements between system refinement and system decomposition levels
- require satisfaction of requirements for system components
- refine the specification of requirements by behavioral models
- plan, organize and log activities for verification and validation of requirements

Requirements vs Model

Requirement entity identify textually defined, compulsory properties

Formalisation of requirements with Constraints, behavioural entities, etc. are possible with RefineReq relation

Depending on the roles in a model exchange:

- An entire model may be seen as a requirement
- Selected parts of the model are required
- An entire model may be seen as a specification

Requirements – Idea & Concepts

Req. support in EAST-ADL2 is based on “**Generic Requirements**“

= simple objects that contain all information (except ID and text. description) in customized attributes (same philosophy as DOORS and RIF) + support for links and groups

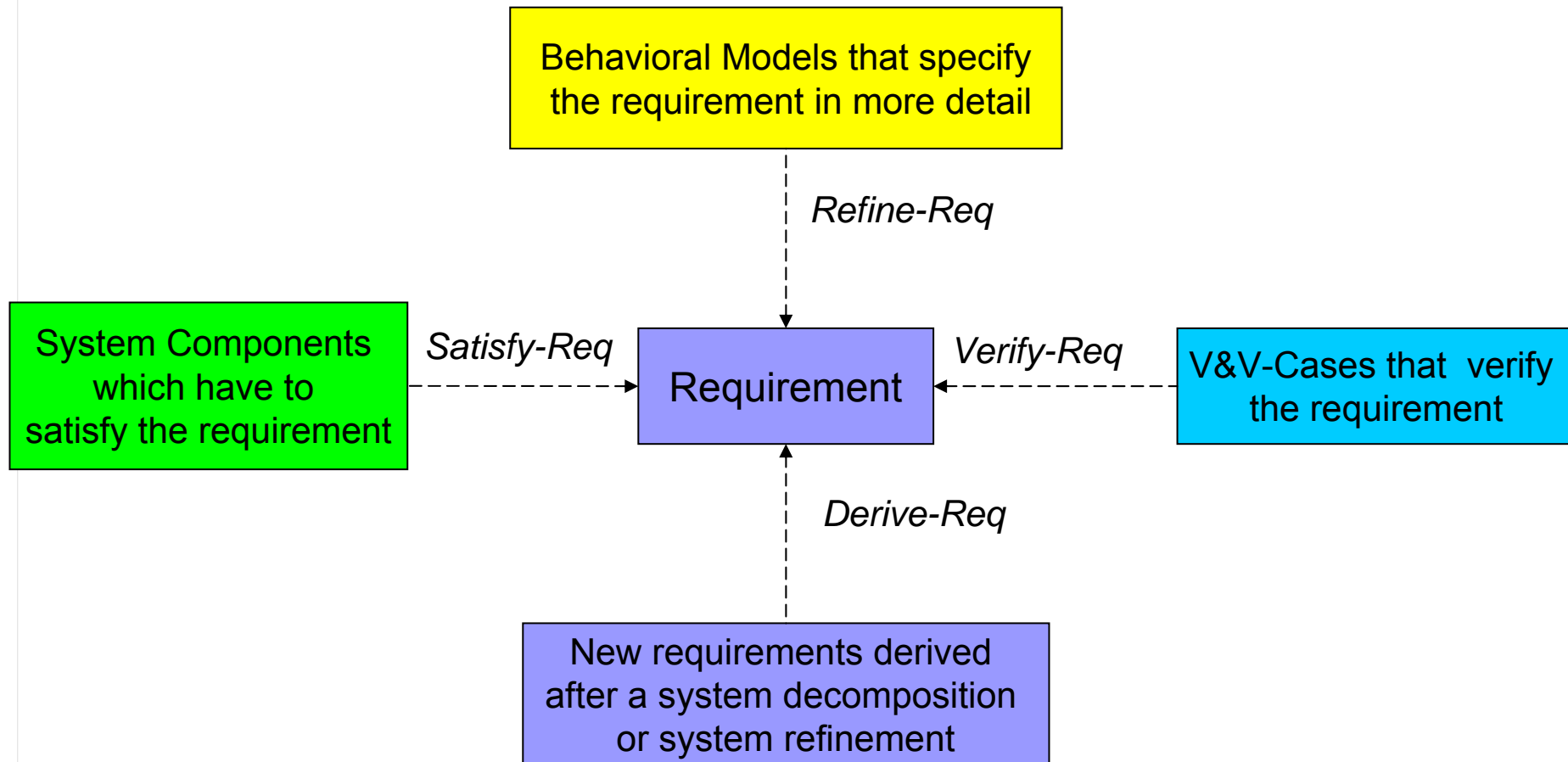
New concept for the customized attributes is called “**User Attributes**“

User Attributes can then be made available for other EAST-ADL2 elements as well (e.g. ADLFunctionType) → they become concept for overall project-/company-specific customization of the EAST-ADL2

In addition to generic requirements: **Specialized Requirements** add certain attributes and associations for a special purpose and with a special semantic (e.g. timing requirements)

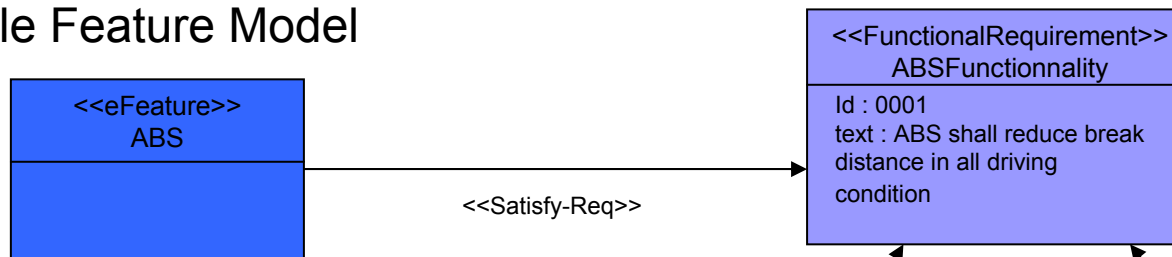
→ allow for tight coupling with system definition (FAA, FDA, HA, etc.)

Requirements – Basic Relations

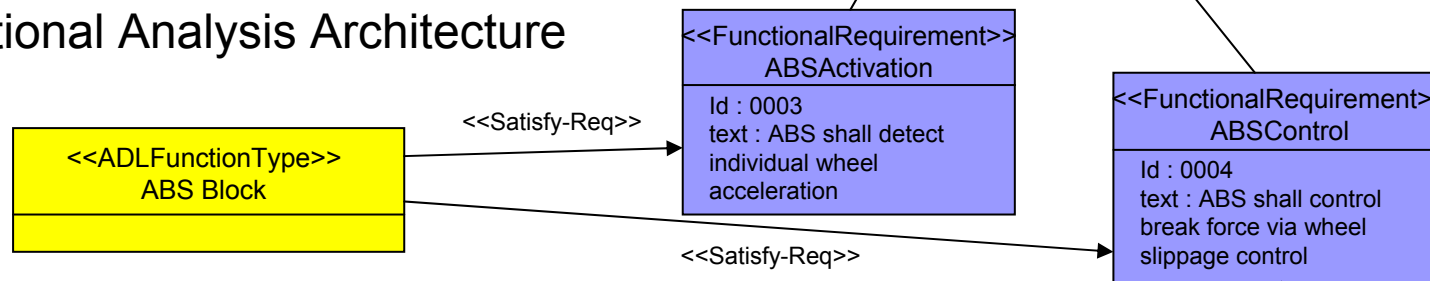


Requirements – Tracing and Linking to system components

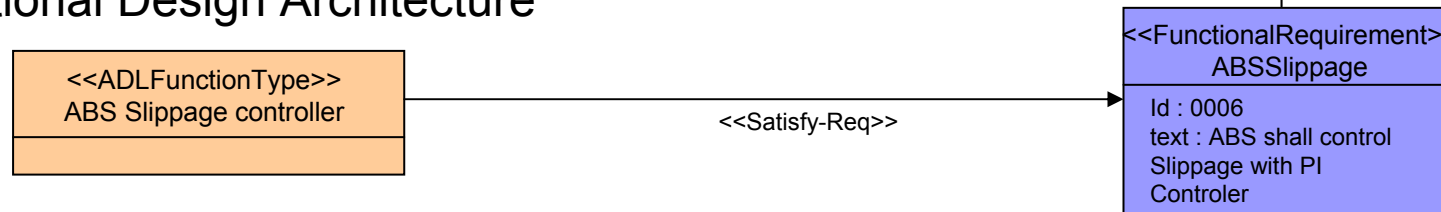
Vehicle Feature Model



Functional Analysis Architecture



Functional Design Architecture

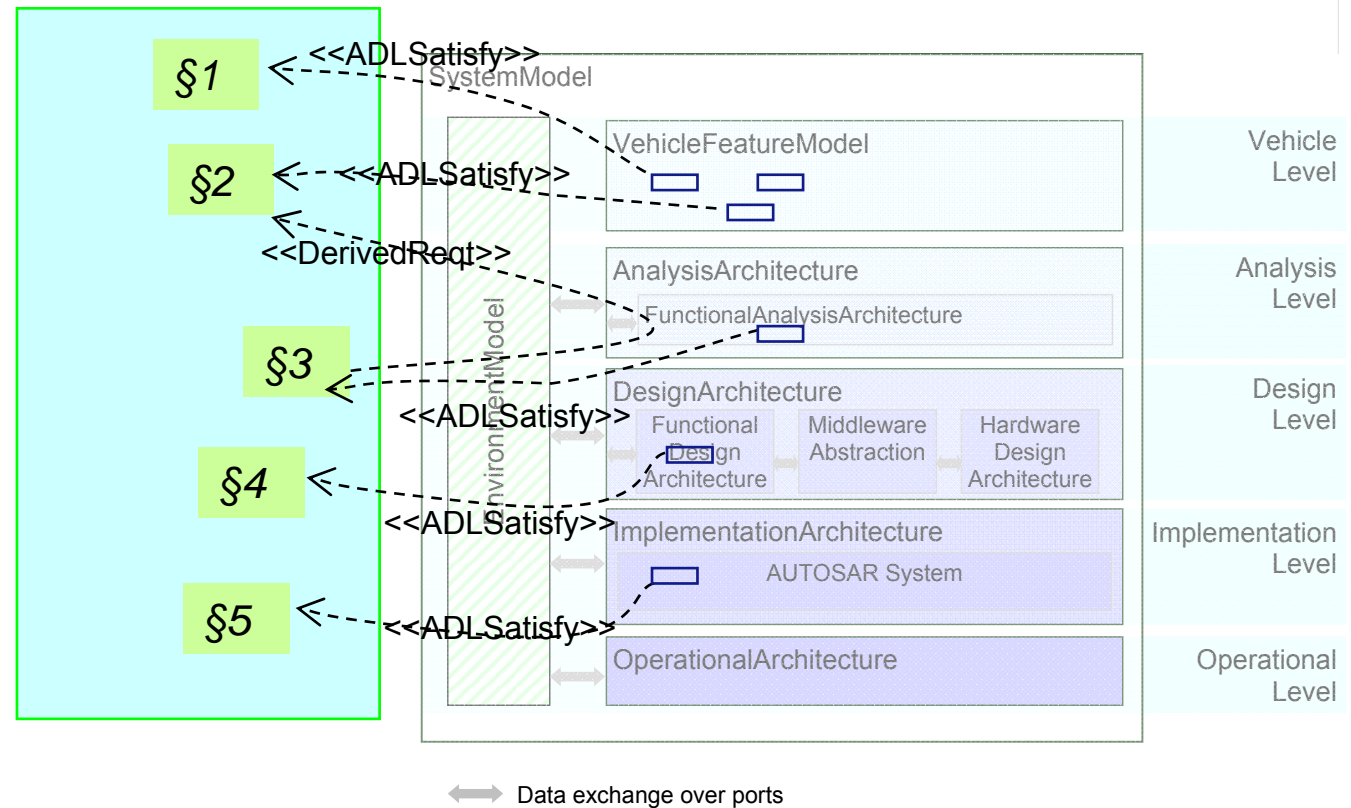


Requirements

Requirements are normally defined in a central repository.

ADLSatisfy relates requirements to features functions and components.

DeriveReq relates a derived requirement to its original.

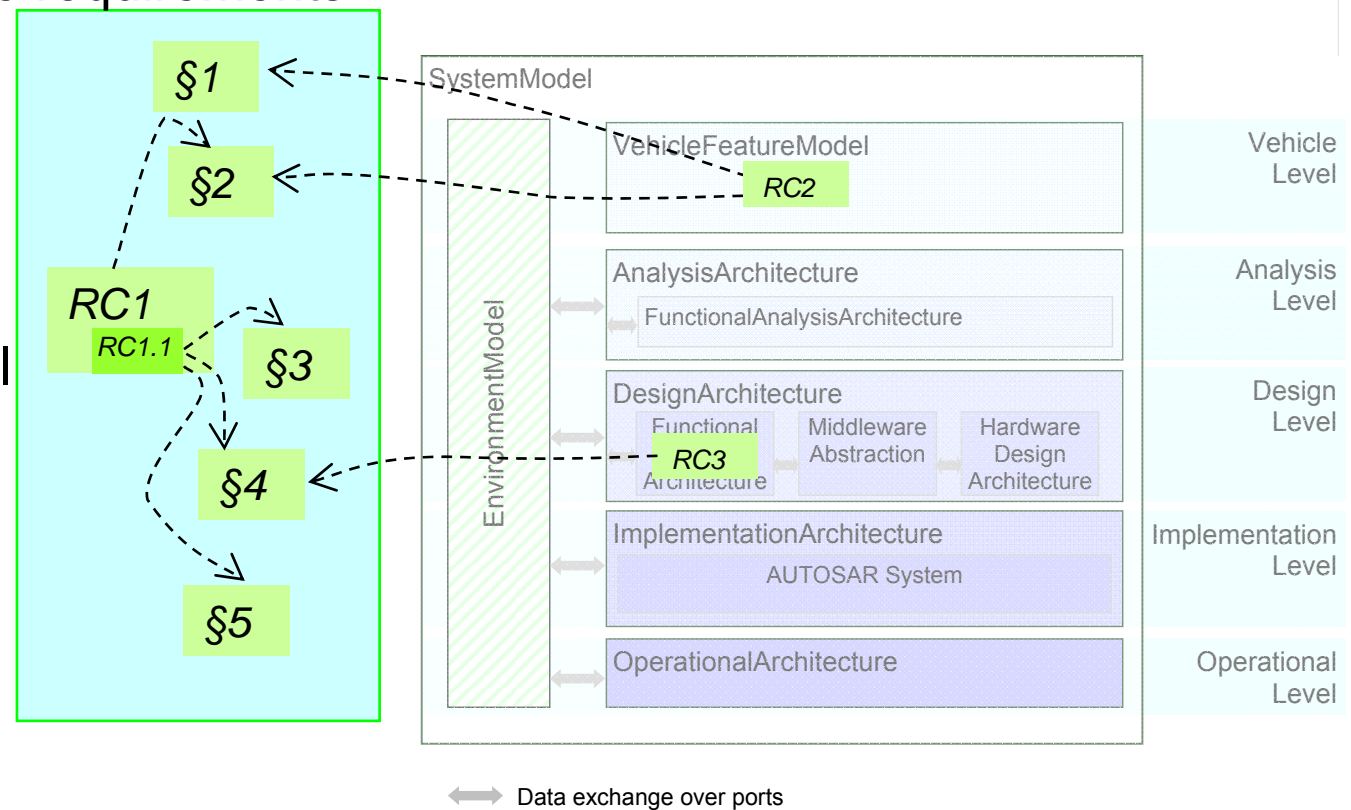


Requirements

RequirementContainers may be used to organize requirements

Containers may be local or global

The same requirement may be linked to several containers.



User Attributes (1/2)

Purpose in EAST-ADL2:

1. providing **generic requirements** with project-specific data
2. augmenting **all elements** of an EAST-ADL2 model with customized, project-specific meta-information

All entities inheriting from metaclass ADLEntity can have user attribute values attached to them.

Scope and structure of this meta-information can be defined on a per-project basis

- by defining user attributes for certain types of EAST-ADL2 elements,
- optionally organized in UATemplates (\approx sets of user attributes).

User Attributes (2/2)

based on two concepts:

1. user attribute values attached to model elements
2. user attribute definitions defined for certain types of elements

mechanism optimized for flexibility and simplicity

- attribute values attached to an element may well conflict the attribute definitions in effect for this element
- attribute definitions meant as (1) guideline for the engineer and (2) as a basis for *optionally* checking validity of attribute values

⇒

- avoids complex interdependencies between parts of the model
- many intricate situations during creation and evolution of a model are circumvented

V&V Support – Basic Concepts

VVCase = a certain, overall V&V effort of varying scope and intention

- core concept of V&V support in EAST-ADL2
- e.g. safety analysis ; specification, design or implementation review ; analysis or design level simulation, SIL-testing, HIL-testing, or vehicle testing

VVProcedure = individual task in the context of an overall V&V effort (i.e. a VVCase), which has to be performed in order to achieve that effort's overall objective.

VVTarget = concrete testing environment in/on which a particular V&V activity (i.e. VVProcedure) can be performed

- can be physical hardware or pure software (e.g. design level simulations)

VVLog = captures outcome of an actual execution of a V&V activity

➔ numerous other, subordinate entities exist
(e.g. VVStimuli, VVIntendedOutcome, VVActualOutcome, ...)