Information Society Technologies

ATESST

- “Advancing Traffic Efficiency and Safety through Software Technology”

Open Workshop –

WT2.1: Requirements, V&V





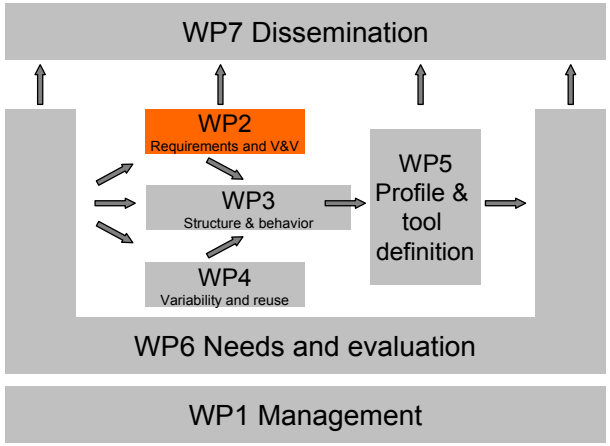


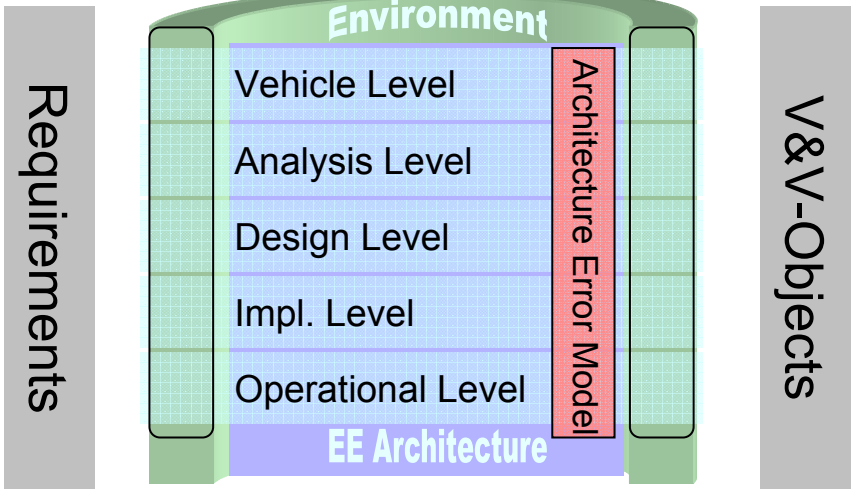




WP2 – Overview and Goals

- General modeling support for requirements and verification and validation (V&V) concepts
  ➔ WT2.1





- Modeling support for safety-analysis techniques
  ➔ WT2.2

WT 2.1 – Goals

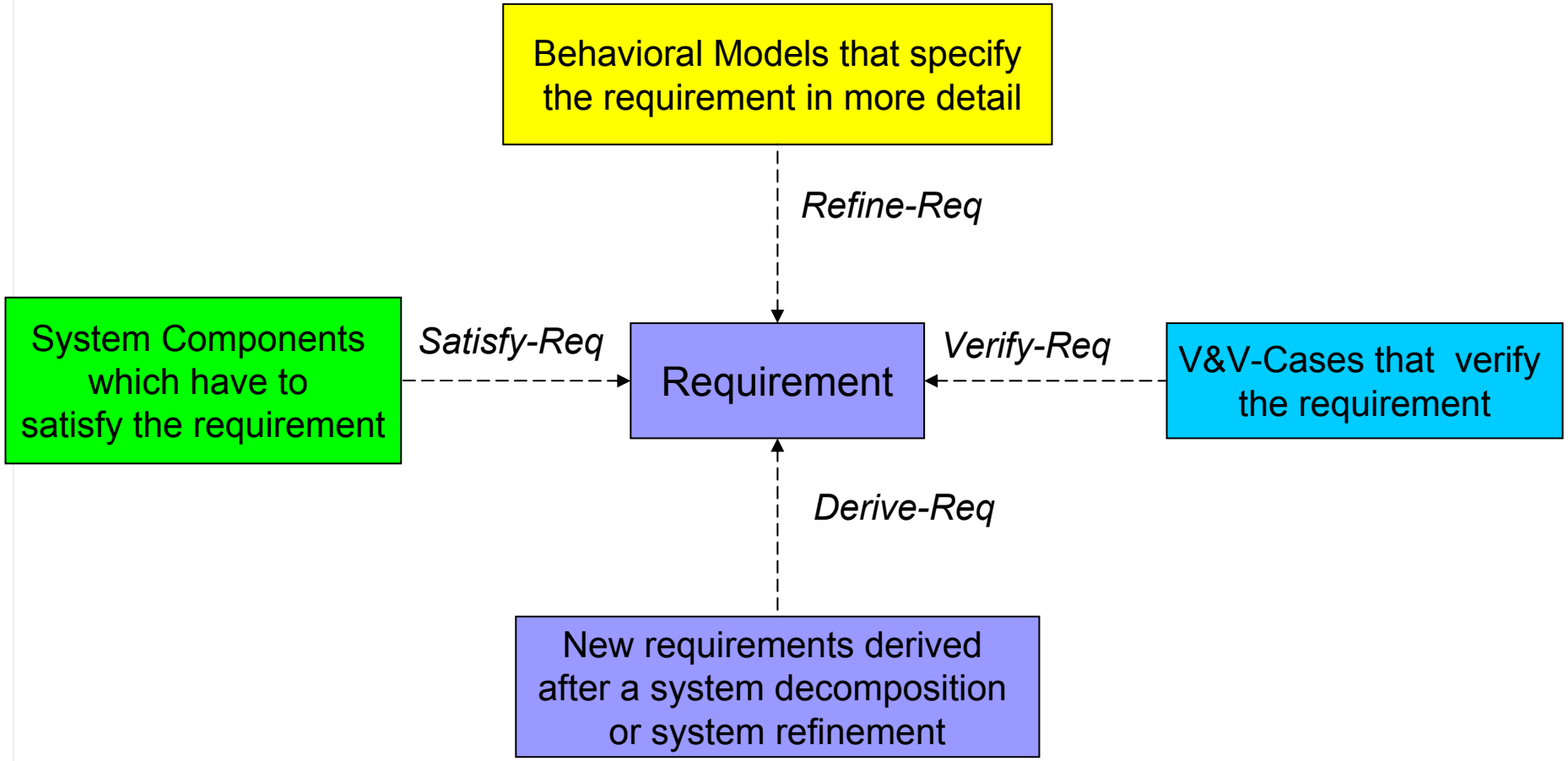
To provide language means to

- 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

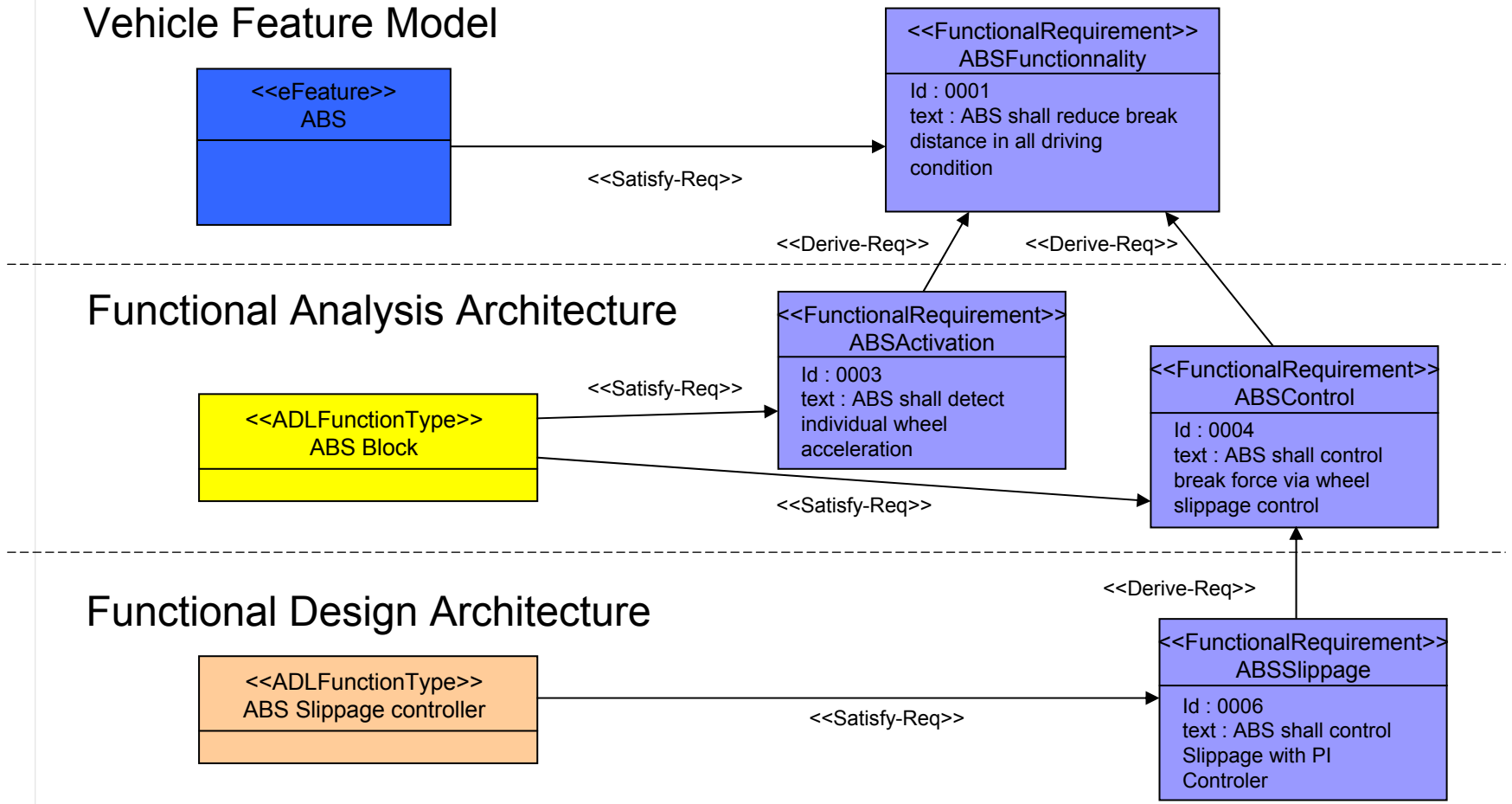
Requirements – Idea & Concepts

- 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



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 (≈ 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

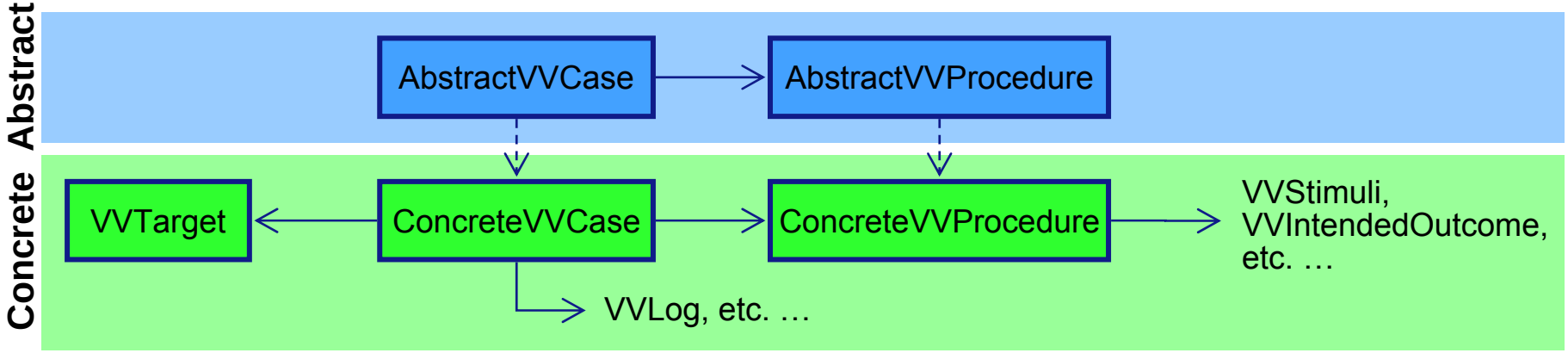
V&V Support – Basic Concepts

- **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, …)

V&V Support – Abstract vs. Concrete

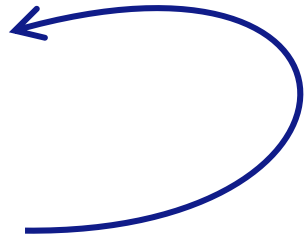
- **Abstract level:** V&V information defined …
  - without referring to a concrete testing environment,
  - without specifying stimuli and the expected outcome of a particular VVProcedure on a detailed technical level.
- **Concrete level:** V&V information here …
  - specifies a concrete testing environment,
  - provides all necessary details for testing (e.g. stimuli and expected outcomes) on a concrete technical level which is applicable to that testing environment.



Tool Support for Requirements

Desired:

1. Editing Requirements
2. RIF Import and Export



provides one possibility for editing (i.e. in external tool)

► *not yet another tool for editing requirements*
<u>instead:</u> import / export facility to make use of legacy tools

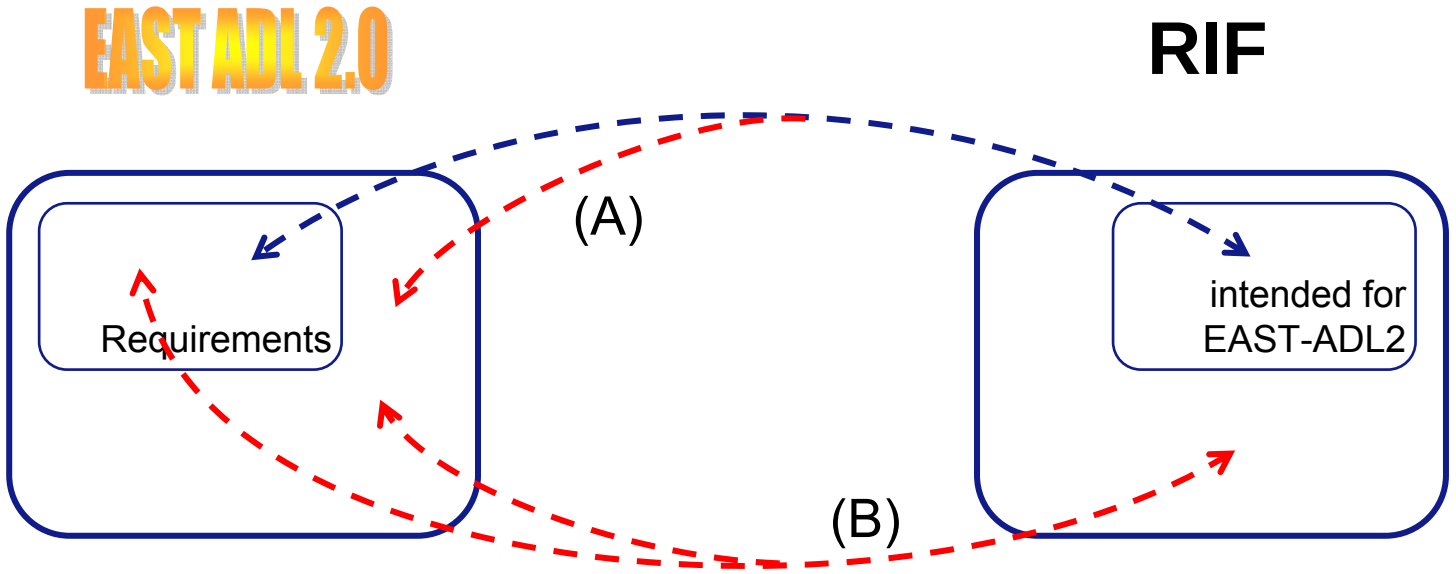
RIF Import/Export – Motivation

Rationale for targeting RIF:

- great flexibility
- open
- standardized
- RIF is becoming increasingly popular
- fits well to decisions taken in EAST-ADL2

  (e.g. a requirement mainly consists of project-specific attributes)

RIF Import/Export – Approach



Important objectives beyond a basic Import / Export:

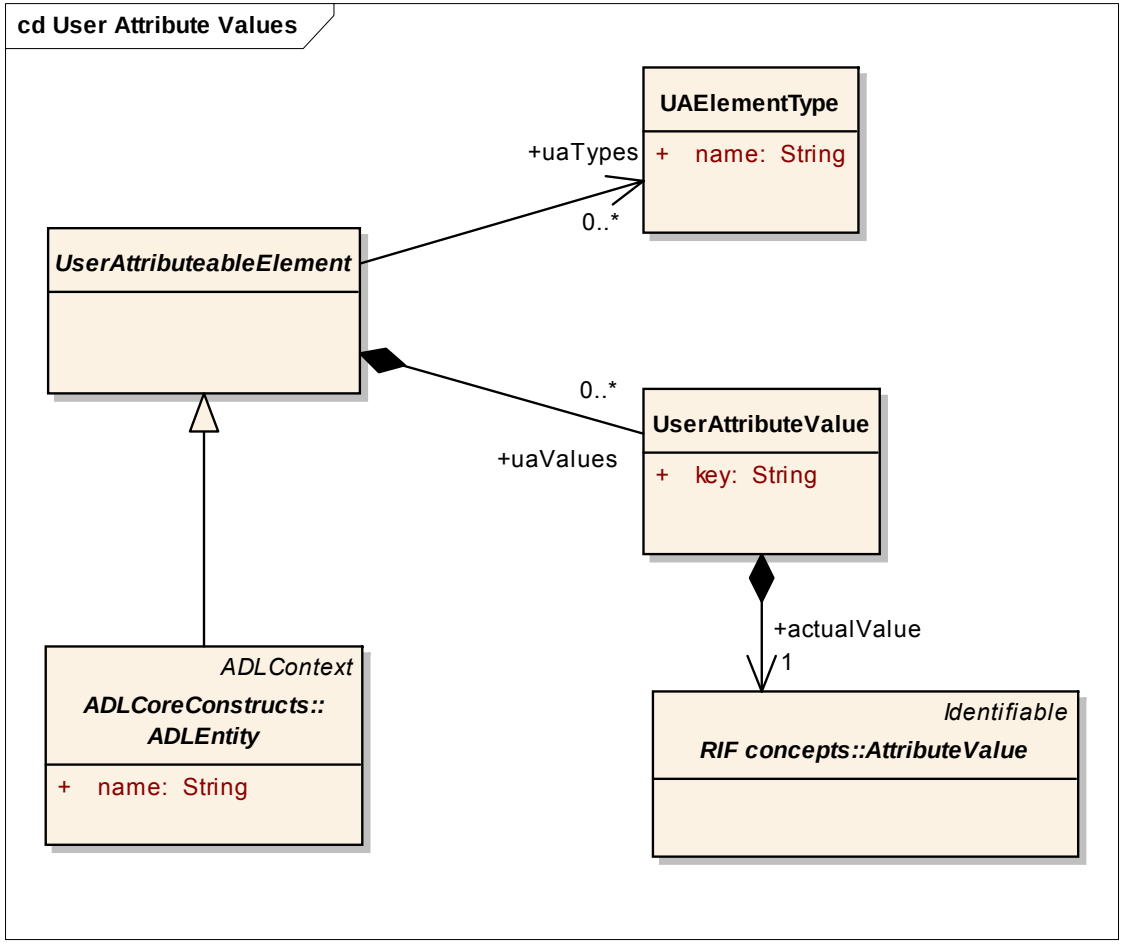
(A) all EAST-ADL2 elements can be target of an import (not only requirements)

(B) support for import of RIF data which was **not** intended for EAST-ADL2

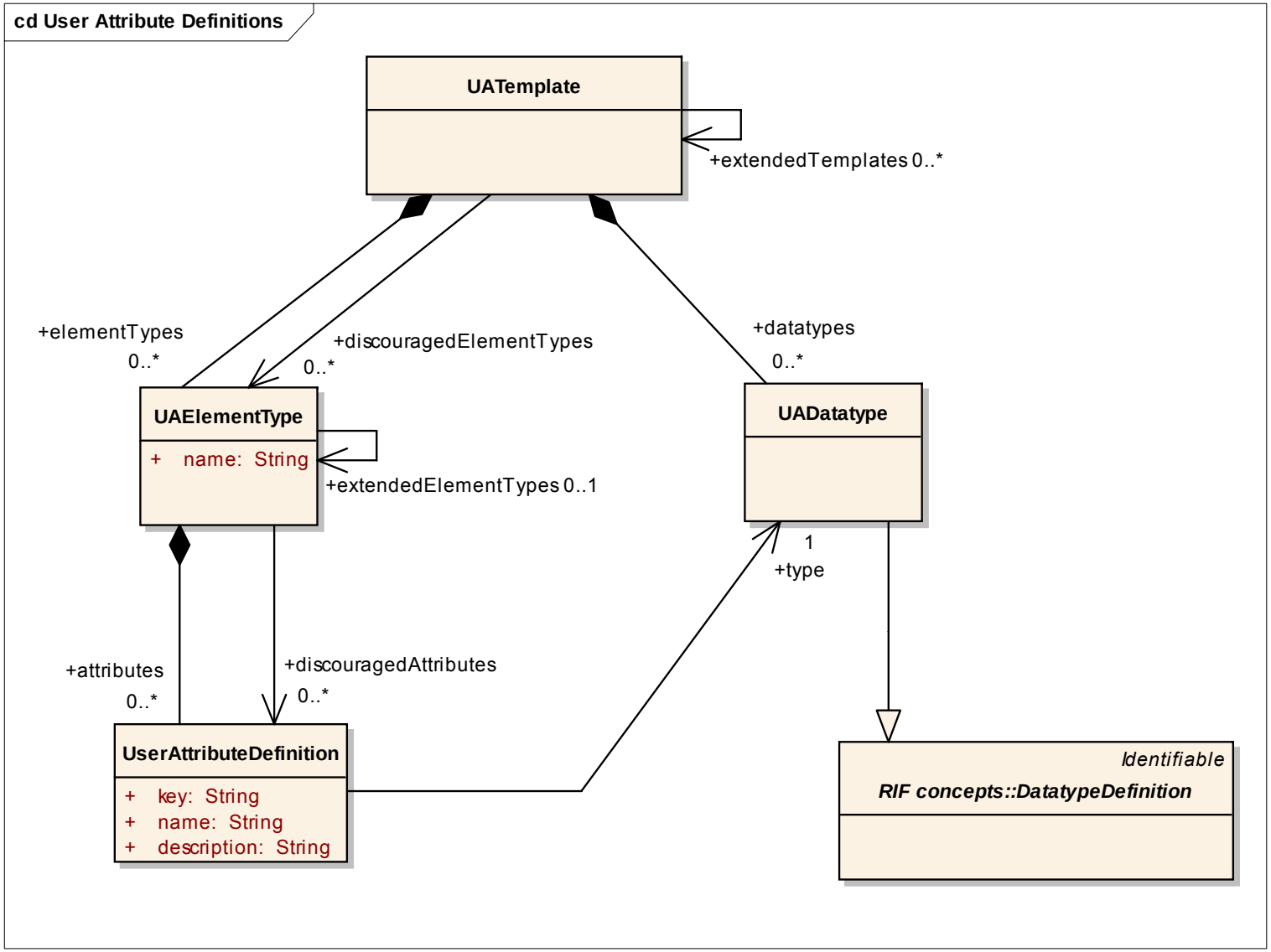
(both apply correspondingly to export)

BACKUP SLIDES

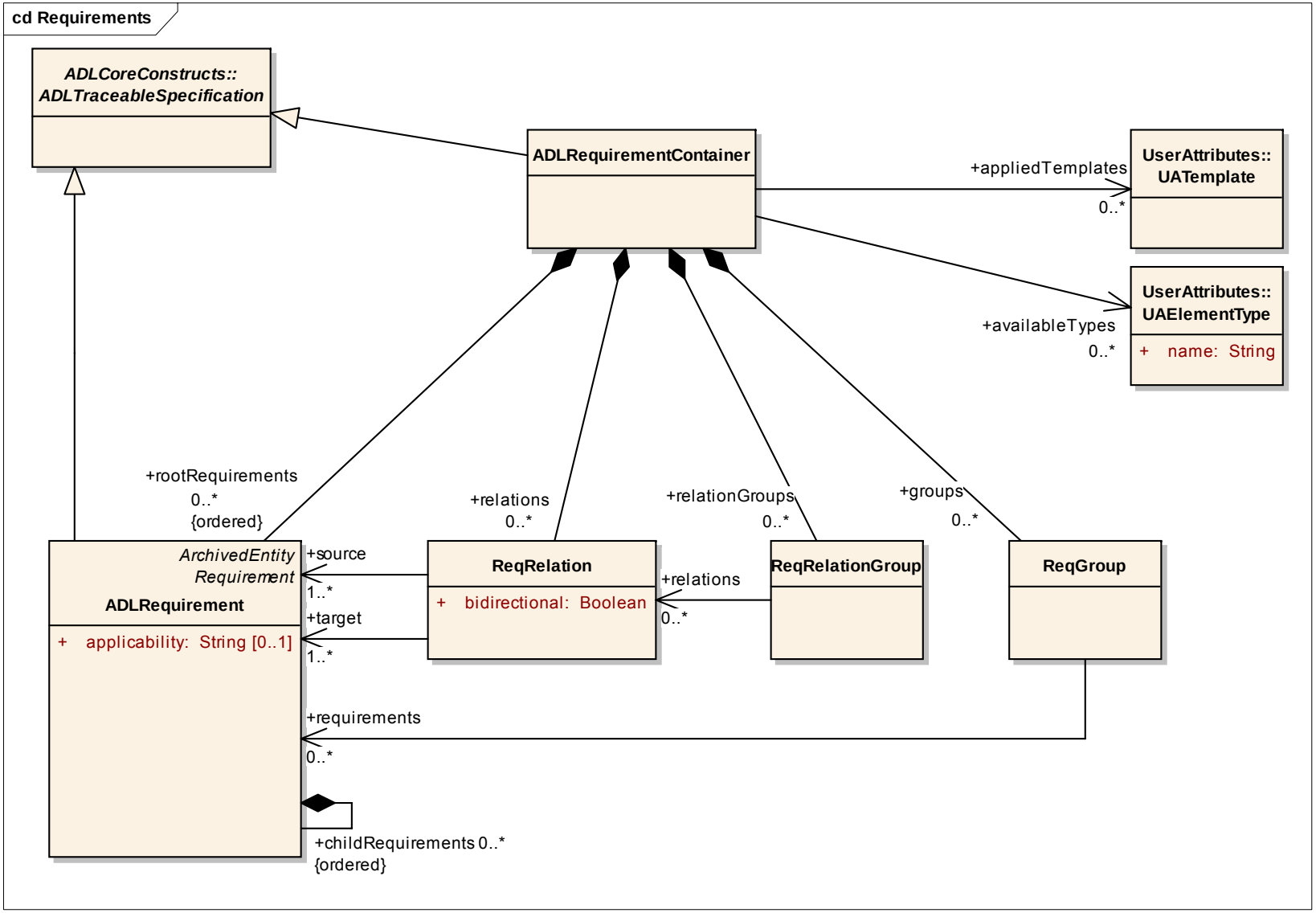
User Attributes - Values

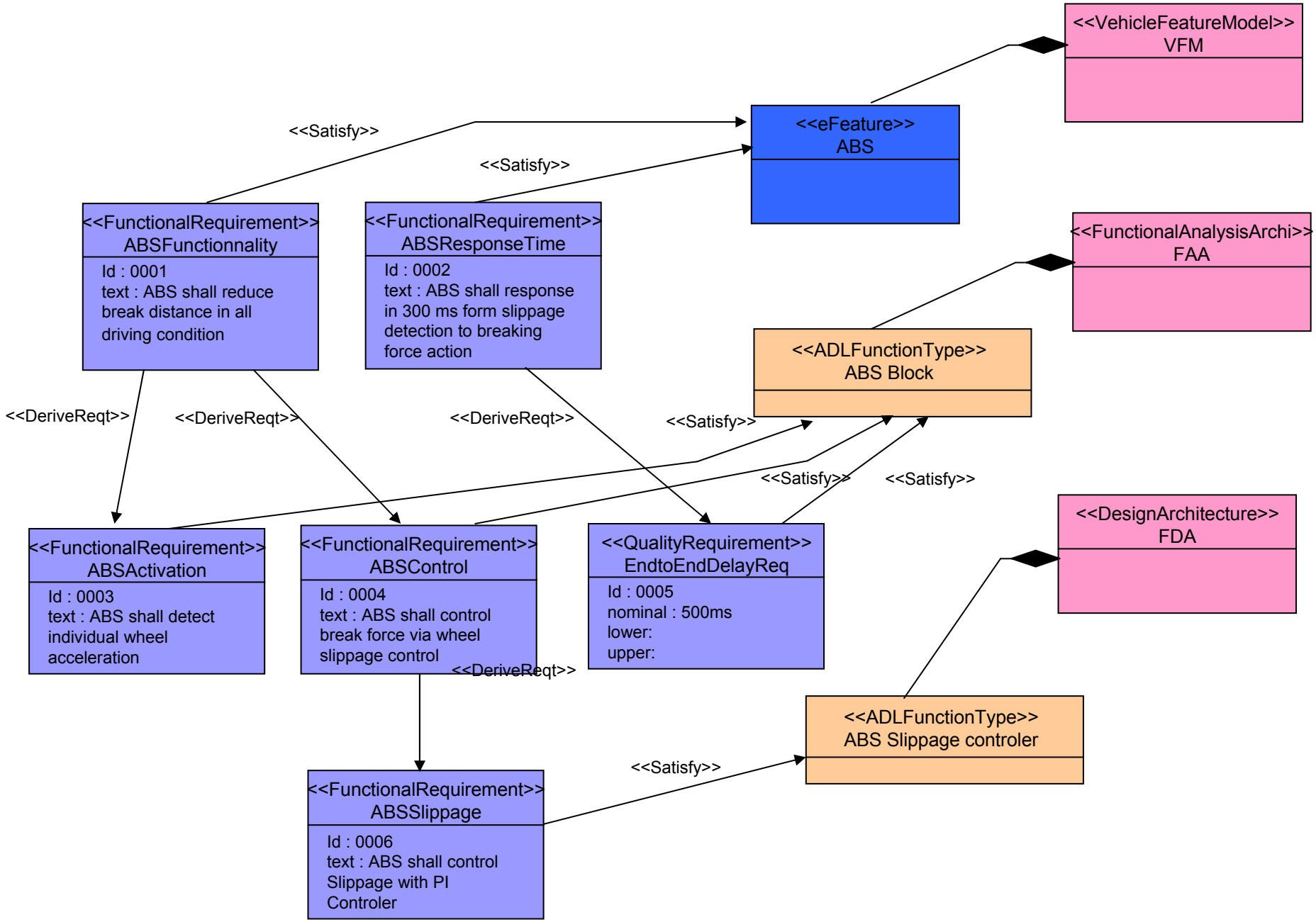


User Attributes - Definitions

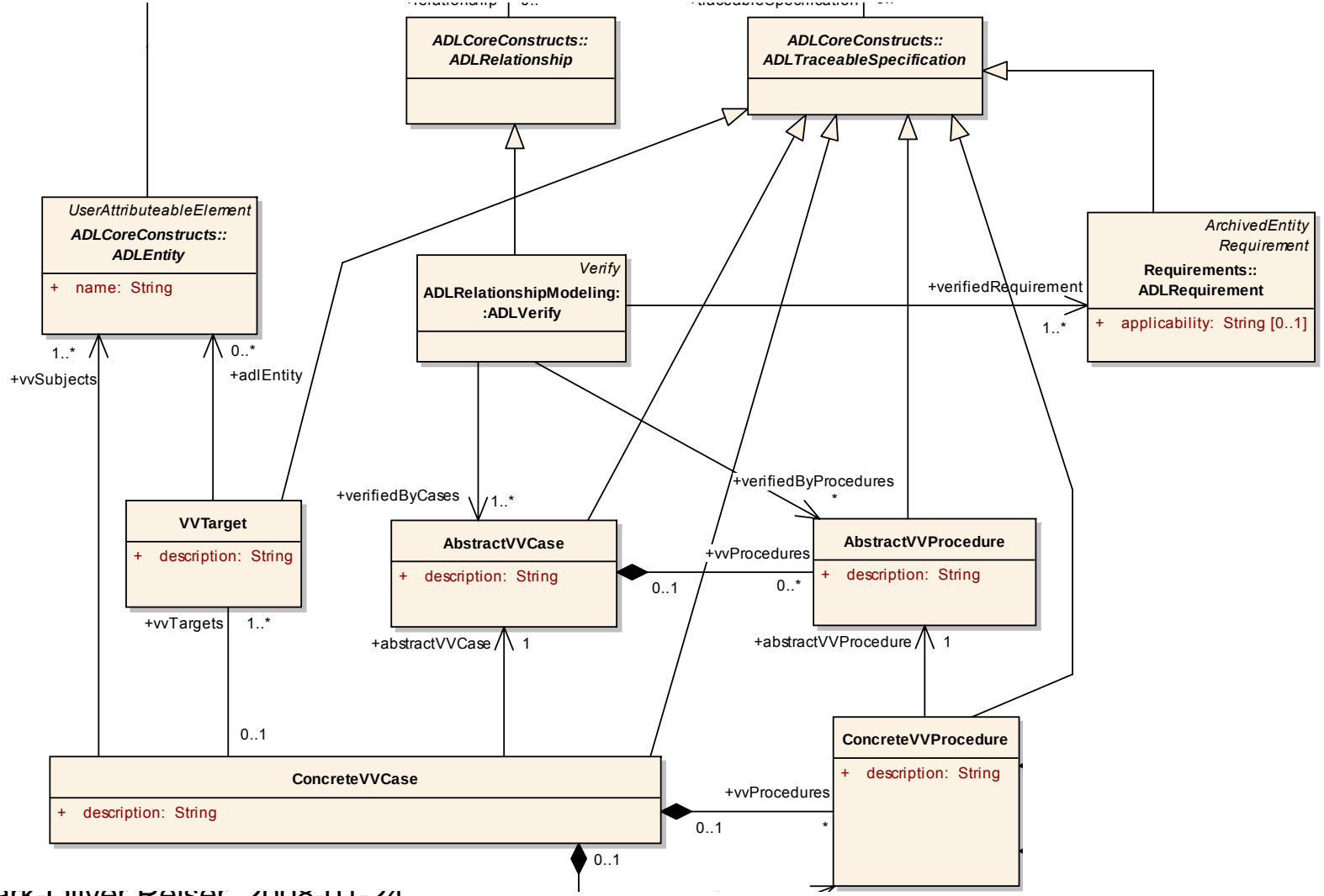


Requirements – Generic




<<VehicleFeatureModel>>
VFM
<<eFeature>>
ABS
<<Satisfy>>
<<Satisfy>>
<<FunctionalRequirement>>
ABSFunctionnality
Id : 0001
text : ABS shall reduce break distance in all driving condition
<<FunctionalRequirement>>
ABSResponseTime
Id : 0002
text : ABS shall response in 300 ms form slippage detection to breaking force action
<<FunctionalAnalysisArchi>>
FAA
<<ADLFunctionType>>
ABS Block
<<DeriveReqt>>
<<DeriveReqt>>
<<DeriveReqt>>
<<Satisfy>>
<<Satisfy>>
<<Satisfy>>
<<FunctionalRequirement>>
ABSActivation
Id : 0003
text : ABS shall detect individual wheel acceleration
<<FunctionalRequirement>>
ABSControl
Id : 0004
text : ABS shall control break force via wheel slippage control
<<QualityRequirement>>
EndtoEndDelayReq
Id : 0005
nominal : 500ms
lower:
upper:
<<DesignArchitecture>>
FDA
<<DeriveReqt>>
<<ADLFunctionType>>
ABS Slippage controler
<<Satisfy>>
<<FunctionalRequirement>>
ABSSlippage
Id : 0006
text : ABS shall control Slippage with PI Controler

Verifiaction & Validation – Part 1/2



Verifiaction & Validation – Part 2/2