EAST-ADL

Concept Presentation

Dependability Analysis

Background

- EU requires a **reduction about of 50 % of the dead** rate due to **road accidents** before **2010** and about **75% before 2020**.
- The **reduction of fatalities will be considered an important goal** supported by: **new passive, preventative and active safety systems** that decrease the probability of an accident and mitigate the consequences of accidents
- **Advanced driver assistance systems (ADAS)** are expected to play a major role in road safety in Europe
- **New functionalities** for **active safety,** to help guarantee **Maximum Vehicle Stability** and to support **Automatic Recovery in Emergency Maneuvers,** are starting to be available on the market
- The automotive industry shares the view that in the next 10 years, 90% of its expected innovations will be based on **Electrical Electronic systems** with a huge emphasis on **Safety Systems**

Context

New Standard

ISO-CD26262

Functional Safety

FIAT Auto, CRF

BMW, Bosch, Continental,

DC, VW, Delphi, PSA, Renault, Siemens, Valeo, Magna, ARC Seibersdorf Research,TRW, Honda, Nissan, Toyota, (Denso, Hitachi)

Volvo Cars, AB Volvo, Mecel, Autoliv

Landrover, MIRA

TRW, GM, Delphi

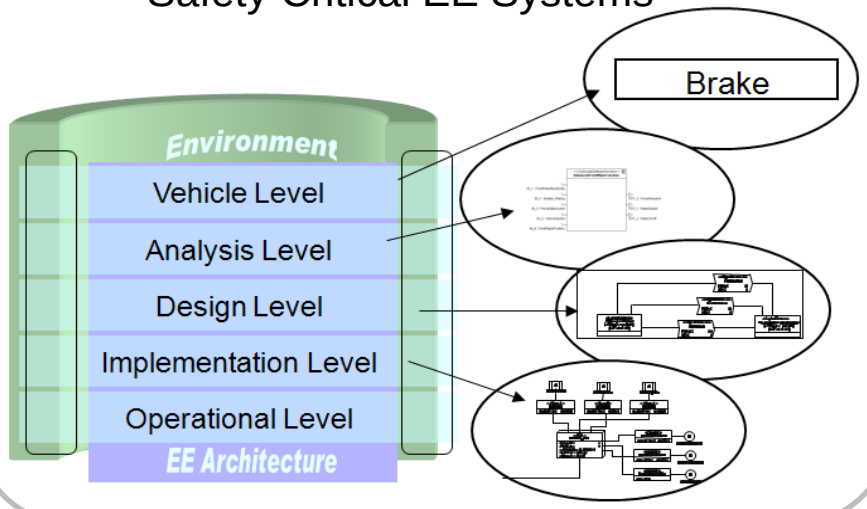
Nissan Europe

Critical Systems Labs

OEMs

Suppliers

Safety Critical EE Systems



Standard Extension

ISO/IEC 15504 SPICE

Software Process Improvement and Capability determination

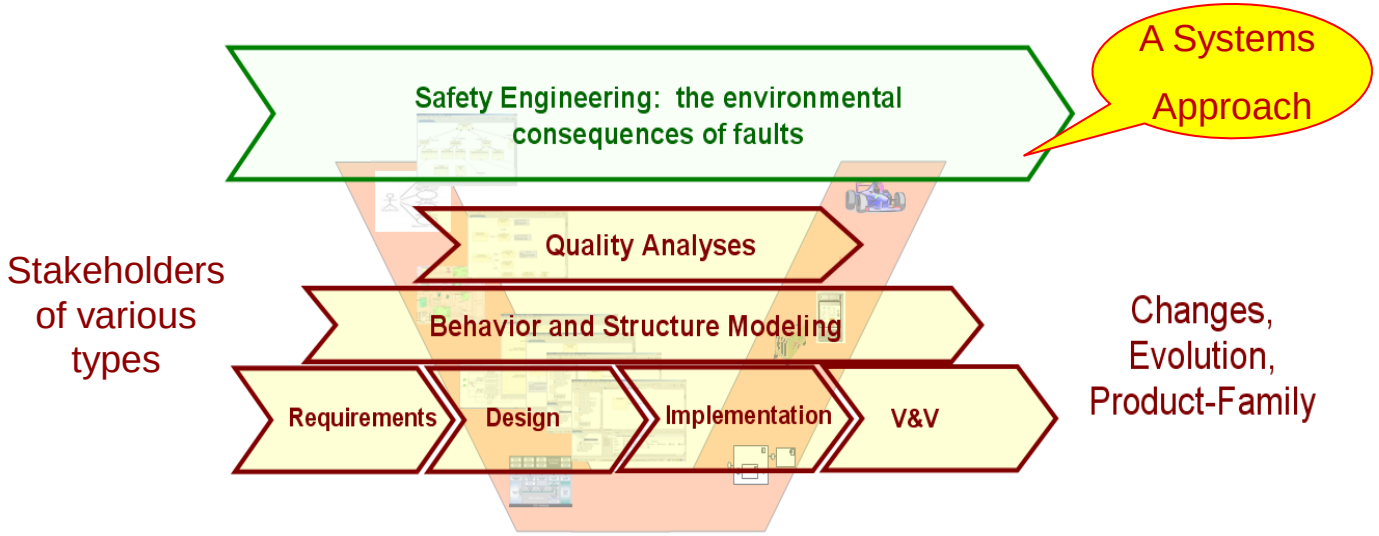
Technology

AUTOSAR With safety extension (BMW , DC, VolksWagen, Toyota, Ford, PSA, FIAT Auto – CRF…)

FLEXRAY With safety extension (BMW , DC, Audi, Volvo, PSA, Renault, FIAT Auto –CRF …)

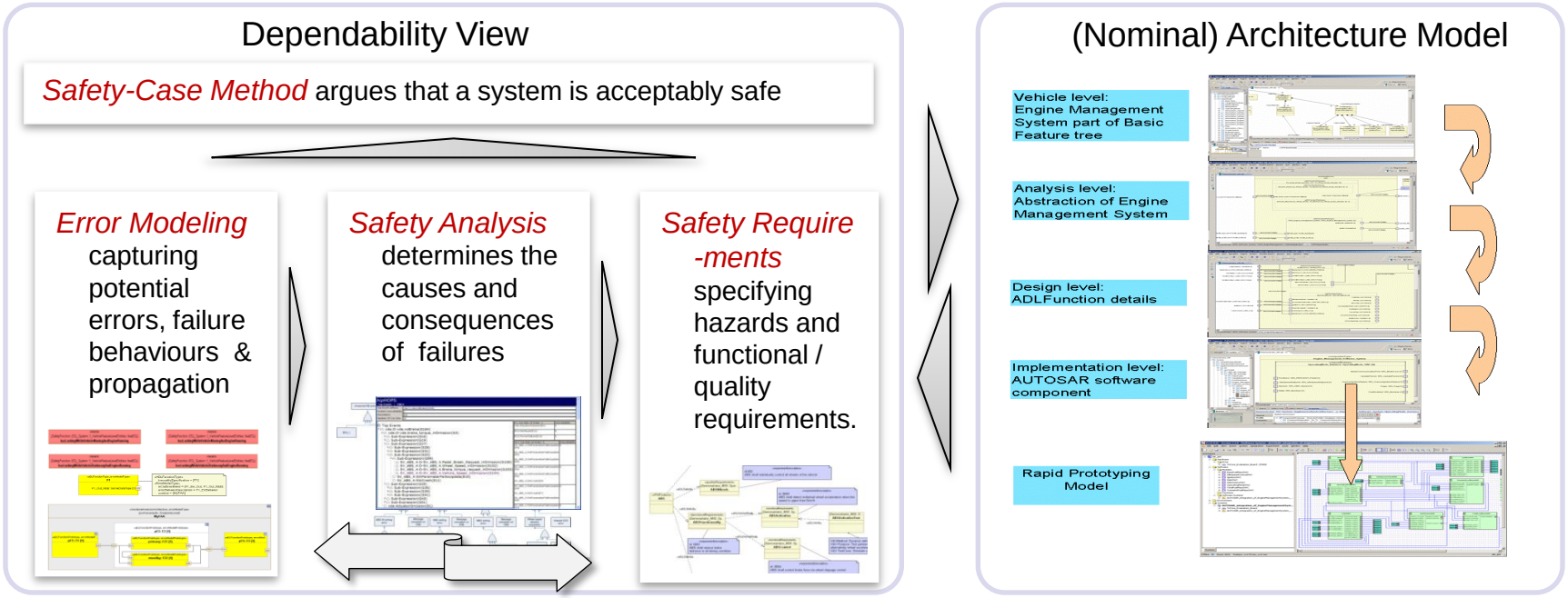
Challenges

- **Difficulties in:**
  - Keeping safety analyses up to date
  - Establishing a complete and consistent view of failure behaviour
  - Managing various analytical information about failure behaviours
  - Proving that a system is acceptably safe in a particular context
  - Avoiding complication of nominal model due to error modelling



EAST-ADL support for Dependability

- **EAST-ADL promotes safety in two ways**
  - Via intrinsic architecture modelling and traceability support
  - Via explicit support for efficient integration of safety engineering activities and nominal architecture design

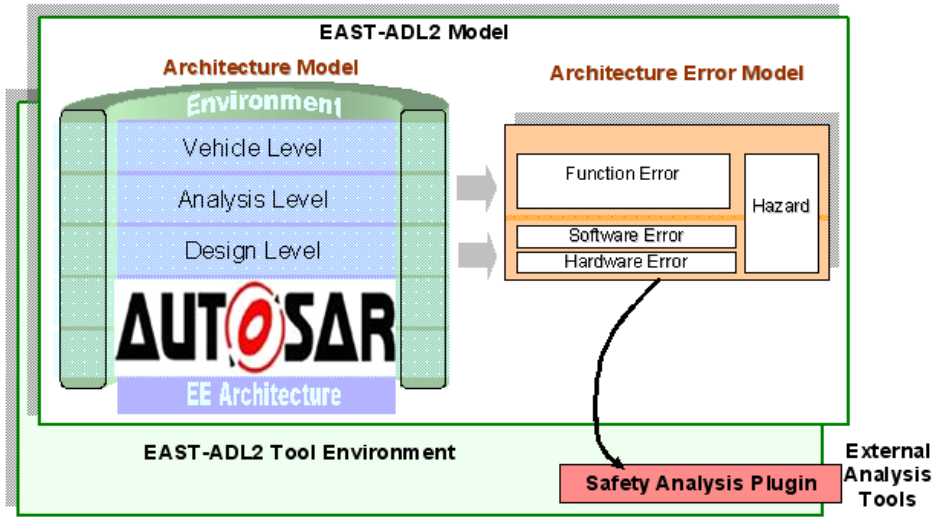


EAST-ADL Dependability Modelling

- **Uses an analytical view that enables:**
  - Explicit modelling of the deviations of functions/systems from their intended behaviour
    - *Extend nominal design with error information*
    - *Exploit semantics of external analysis methods*

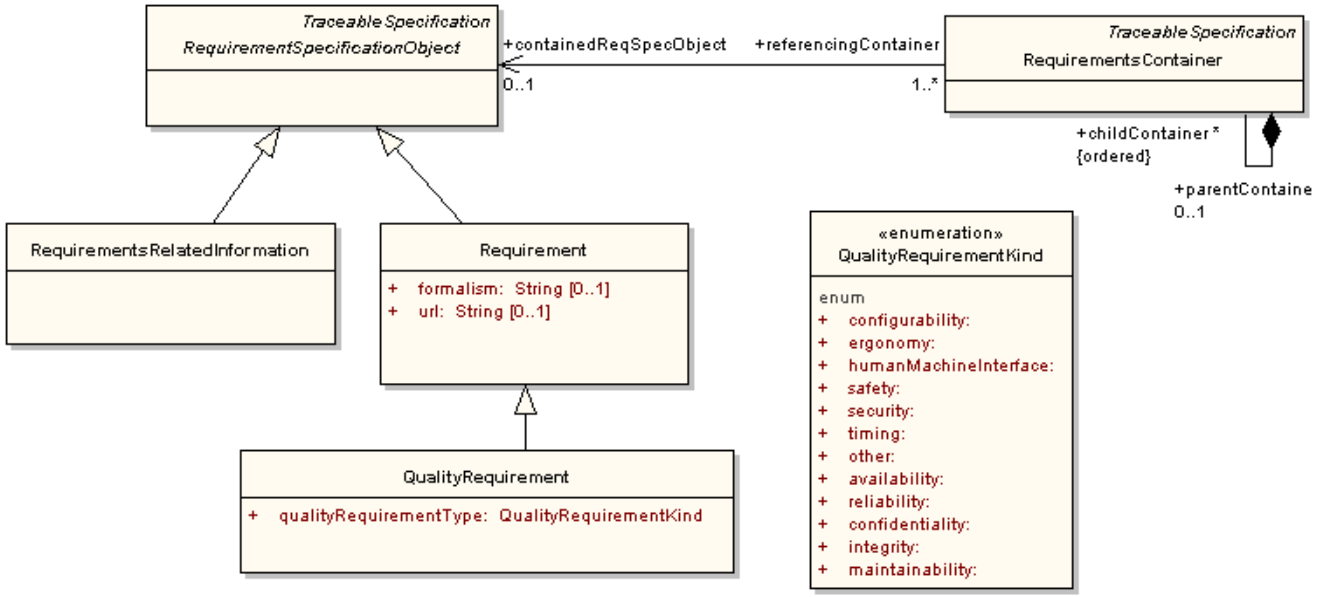
- Seamless integration with architecture development
  - *Traceability of requirements*
  - *Error propagation though architectural relationships*

- Analysis leverage via external tool plugins
  - *Enables assessment of causes and consequences of failures*



Requirements Traceability

- A requirement expresses a condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed property

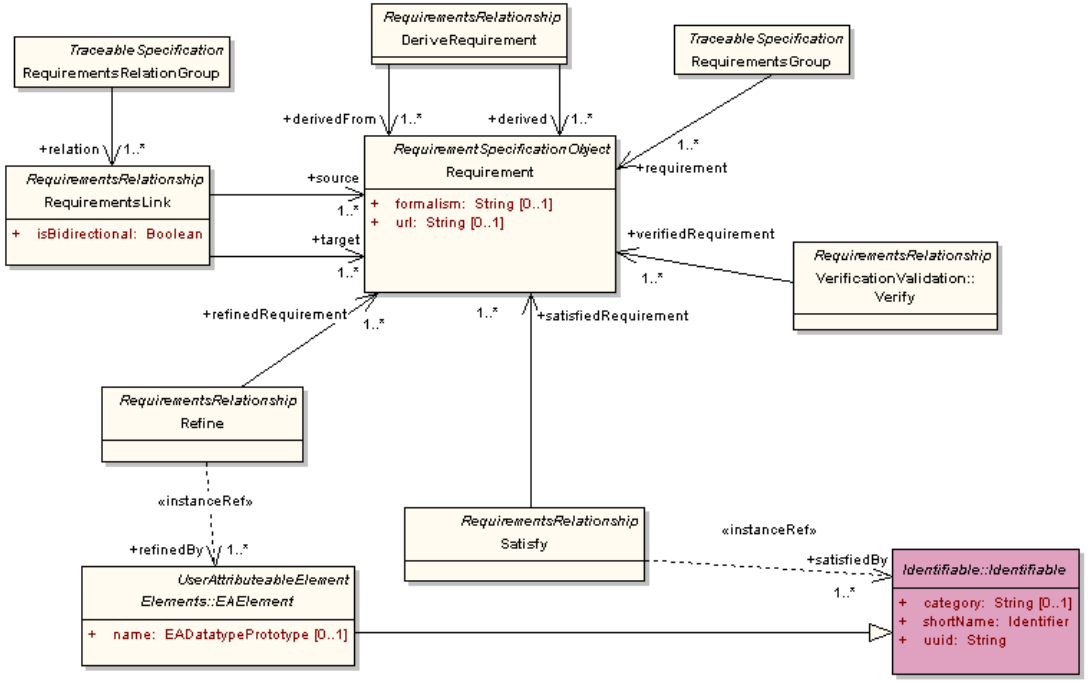


Requirements Traceability

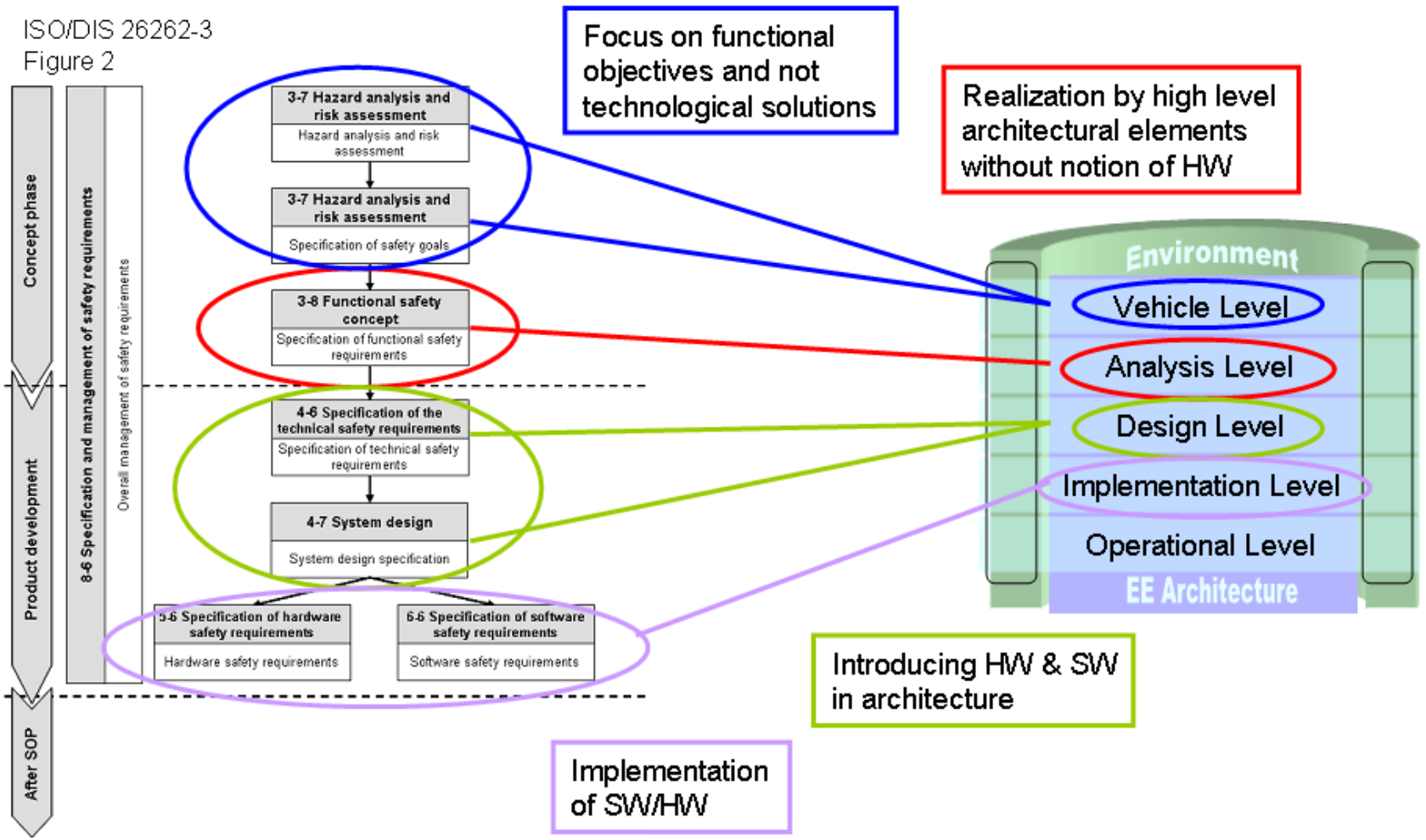
- **EAST-ADL relationships constructs define general purpose relationships to model dependencies between structural constructs**

- The purpose is to formally specify the various relationships that may exist between basic constructs.

- The requirements traceability is modelled in EAST-ADL using these relationships constructs



Abstraction Levels



High abstraction level

- **Definition of the Item**
  - Describe and define the item to develop an adequate understanding of it
- **Initiation of the safety lifecycle**
- **Hazard analysis and risk assessment**
  - Hazardous events are hazards evaluated in an operational situation and are classified with an ASIL value based on severity, controllability, and exposure
- **Functional safety concept**
  - Includes both functional safety requirements (and acceptance criteria) and allocation of functional safety requirements to safety architecture

Hazard analysis & Risk assessment

- Use cases and operational situations define Scenarios
- Safety-oriented use cases may use pre-defined patterns

System name:	name of the System/project Under Discussion (SUD)
Use Case name:	name of the use case
Short description:	short description of the main goals of the use case
Target Function(s:	the function description in terms of output(s) behaviour
Primary actor:	main user of the SUD
Secondary actor(s):	takes advantages from the SUD information but it isn't active into the specific use case
Pre-condition(s):	All the condition to be verified at the beginning of the use case
Application scenario:	application scenario: driving situation (def. WD26262: “scenario that may occur while a vehicle is in use-moving or stationary”) and environmental condition (def. WD26262: “Physical or other constraints under which an item is used”)
Operational scenario:	Sequence of actions and interactions among the system and one or more actors
Fail condition(s):	malfunctions - all different possible termination of the ability of the functionality to perform a function as required
Misuse(s):	incorrect, improper, or careless use of the SUD
Risk’s source:	Origin of the Fail condition/misuse
Function Criticality(s):	Criticality of the function, related to the use case, due to external factor(s)
Post-condition:	describe the condition in which thAe SUD will arrive If the system flow is correct
Status:	description of the use case status (to be approved, approved, in modification,...)
Open issues:	any issues which require discussion affecting this use case
Comments:	any comments on the contents of the use case.

Hazard analysis & Risk assessment

- FeatureFlaw denotes an abstract failure of a set of items
  - i.e. an inability to fulfil one of its requirements
  - Could be due to anomalies or malfunctions of system outputs
  - Or erroneous interaction between systems
- Hazards then represent a system state that may contribute to accidents caused by a FeatureFlaw
- When a Hazard arises in a particular Scenario, it gives rise to a Hazardous Event
  - Represents the effect of that hazard in a particular operational scenario
- Hazardous Events are assigned ASIL values

Hazard analysis & Risk assessment

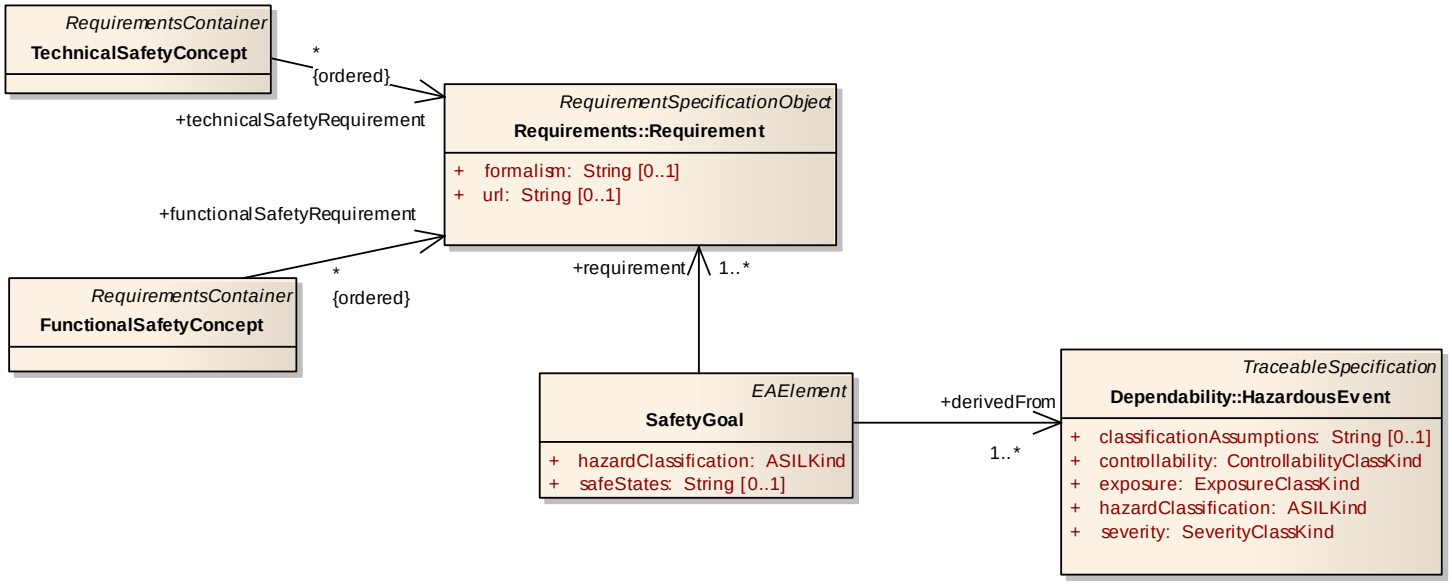
- Safety goals are defined to serve as top-level functional safety requirements
- Purpose of a safety goal is to avoid unacceptable risk posed by hazardous events
  - Should be at least one safety goal per hazardous event
- Each safety goal should have a corresponding Safe State
  - Examples from an electronic steering column lock:
  - Locking should only take place when the conditions are correct
    - Safe states: LockPowerState = Safe Power or Unpowered
  - The reported lock state should always be correct
    - Safe states: LockBoltState = Unknown

Functional Safety Concept

- Represents the set of functional safety requirements allocated to the architectural elements that fulfil one or more safety goals
- Each safety requirement may include:
  - ASIL – a Safety Constraint associated with the requirement
  - Operating Modes
  - Fault Tolerant Time Spans
  - Safe States
  - Emergency Operating Times
  - Functional Redundancies
  - Specifications on how fault tolerance is achieved
  - Acceptance criteria

Technical Safety Concept

- Contains the technical safety requirements
- Details the functional safety concept in the context of the architectural design



Error Modelling in EAST-ADL

- Connection between error modelling and system modelling supports:
  - Quick safety design iterations
  - The creation of dedicated views
  - Structured information management
- Provides structured information handling of:
  - requirements, design, safety analysis, verification and validation information, and design decisions
- Allows reuse, consistency check between models, automated handling of dependencies, view generation, transformations and analysis

Error Modelling in EAST-ADL

- Major error modelling elements include:
  - **ErrorModelType** specifies possible behaviours of a target architectural entity that are of concern when analysing system anomalies and errors
  - **FailureOutPorts** represent a propagation point for failures that propagate out from an ErrorModelType
  - **FaultInPorts** represent a propagation point for faults that propagate into the containing ErrorModelType
  - **FaultFailures** represent internal failures or faults of an ErrorModelType
  - **FaultFailurePropagationLinks** connect multiple ErrorModelTypes together via their ports

Failure logic

- EAST-ADL is tool agnostic and allows different representations of failure logic
- One example is the failure logic used by the HiP-HOPS safety analysis tool
  - Set of logical expressions that link a particular output deviation to a combination (using AND and OR gates) of input deviations and internal failures
  - Uses failure classes to distinguish different types of input/output failure
  - e.g. Omission-Output = Omission-Input OR InternalFailureMode
- This approach allows external tools (like HiP-HOPS) to perform analysis of EAST-ADL error models

Verification and Validation

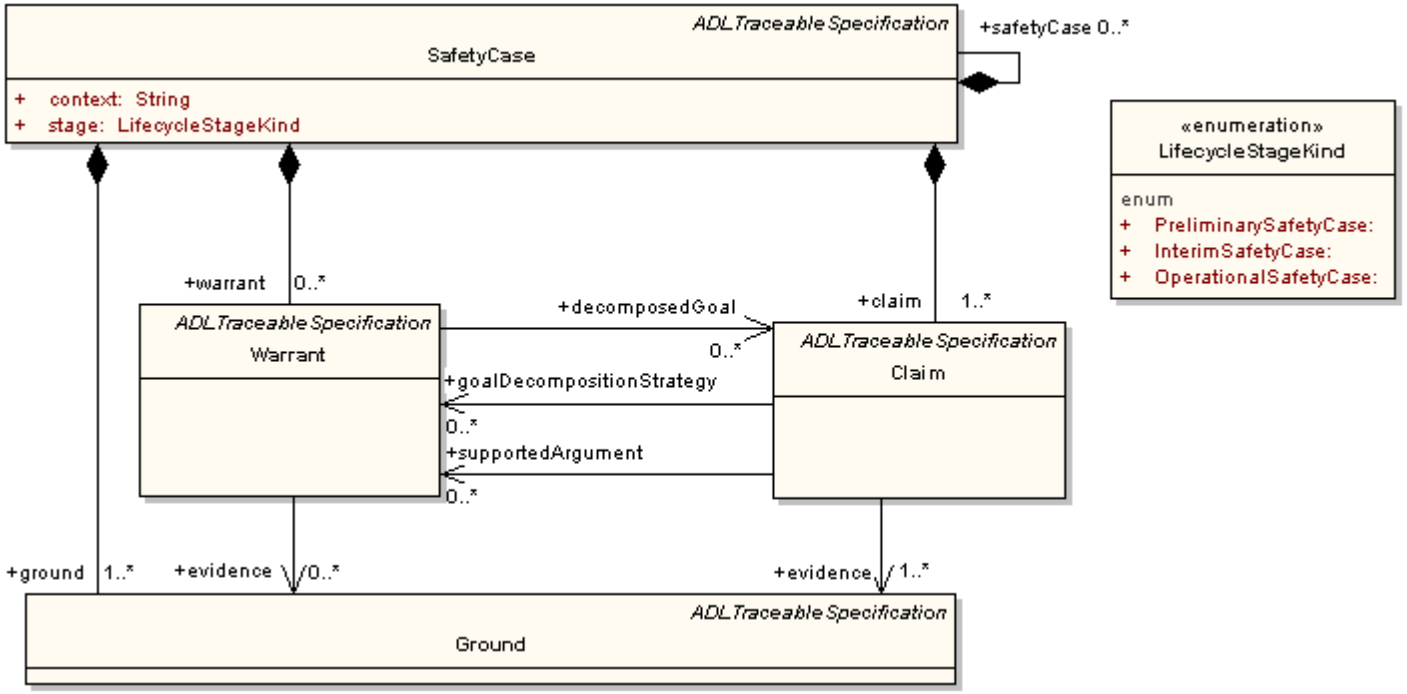
- EAST-ADL provides the means for organising V&V activities on an abstract level
  - Defining the links between V&V activities
  - Defining the requirements that are checked by those activities
  - Defining the objects modelling the system (components, tasks etc)
- Common parts of all V&V techniques are covered by EAST-ADL
  - Expected results from V&V activities
  - Actual results obtained
  - How the V&V activities were constrained
- Information specific to particular V&V techniques is able to be stored but not explicity represented

Safety Case metamodel support

- Structured information management can be used as part of a safety argument in a safety case and supports systematic safety/reliability analysis
- EAST-ADL's support for safety cases addresses an expanding area of functionality with high complexity
- Traceability between safety case and design information facilitates the job of the safety engineer
- Also facilitates the development of safety critical systems and allows impact analysis of elements linked in the safety argument

Safety Arguments in EAST-ADL

- Claim/Warrant/Ground provides means to argue that the development of vehicle systems has been done according to safety norm



Conclusion

- **EAST-ADL provides support for dependability modelling in three important respects:**
  - System development based on models on different levels of abstraction, enabling the fulfilment of many requirements specified by ISO 26262
  - Safety case development in close connection with the design
  - Analysis of hazardous failures by modelling of error propagation in a hierarchical system model
- Integration of these aspects provides structured information handling for requirements, analysis, V&V, and design decisions
- Allows reuse, consistency checks, automated dependency handling, view generation, transformations, and analysis
- Supports safety case development and fast, efficient safety design iterations