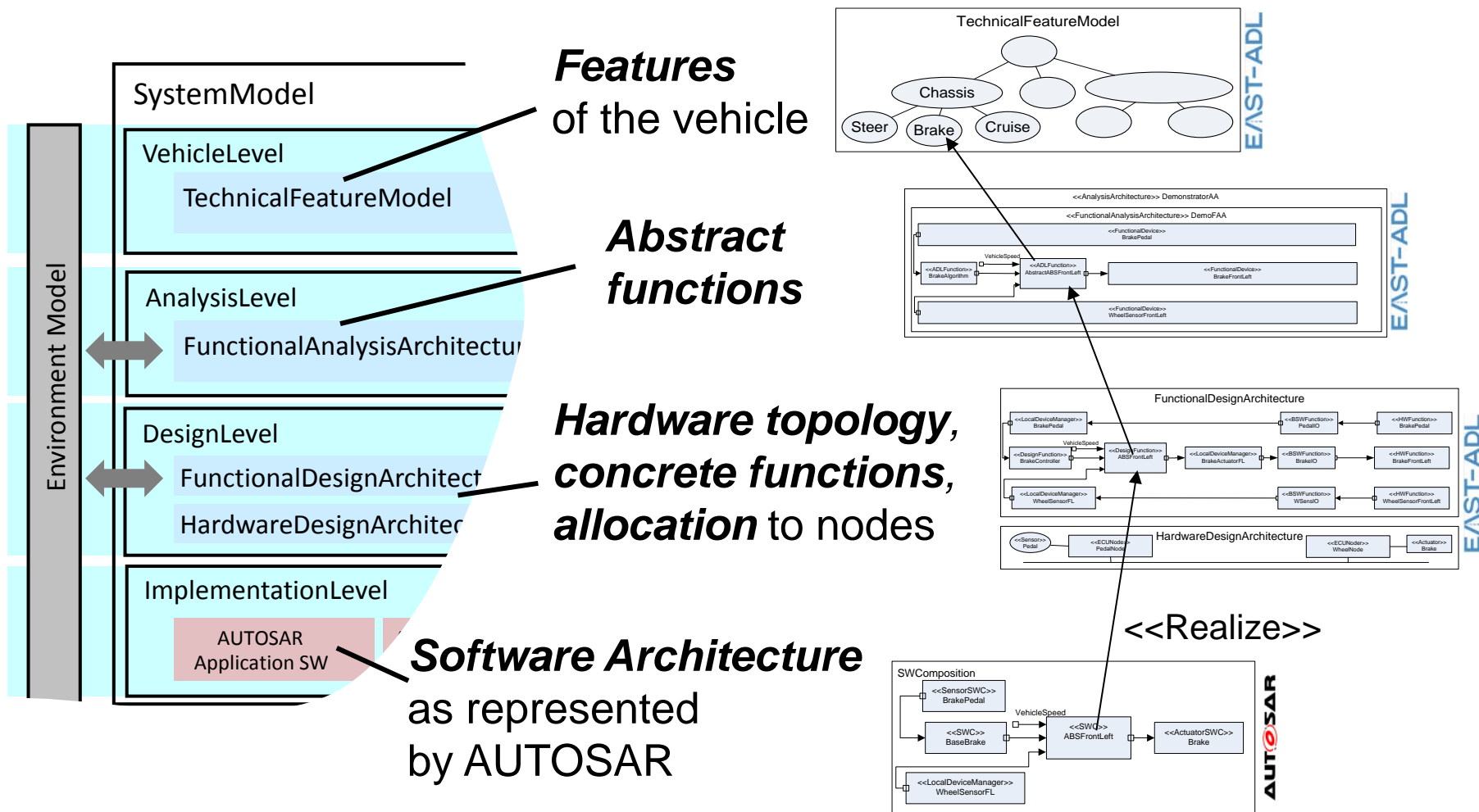
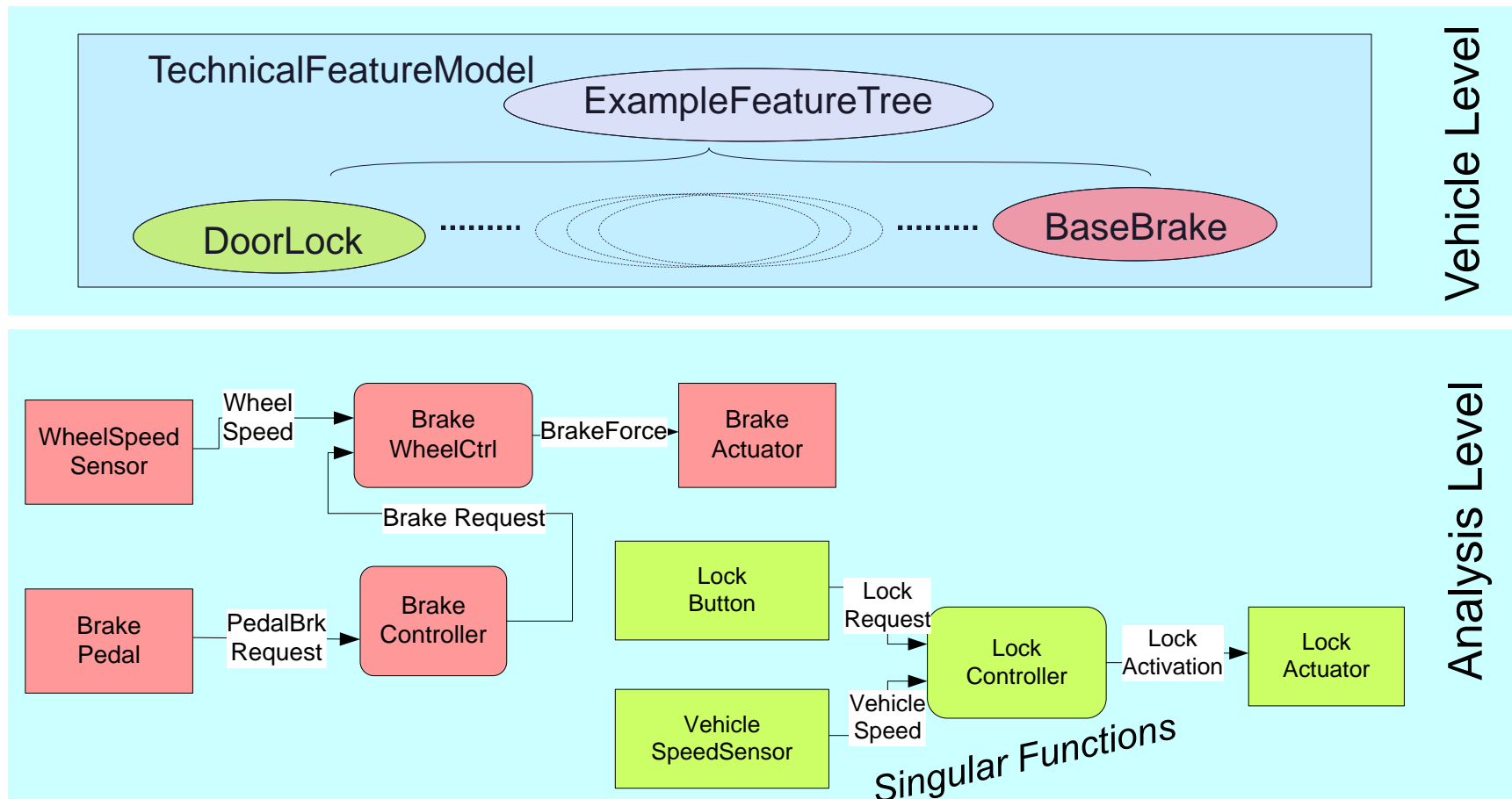


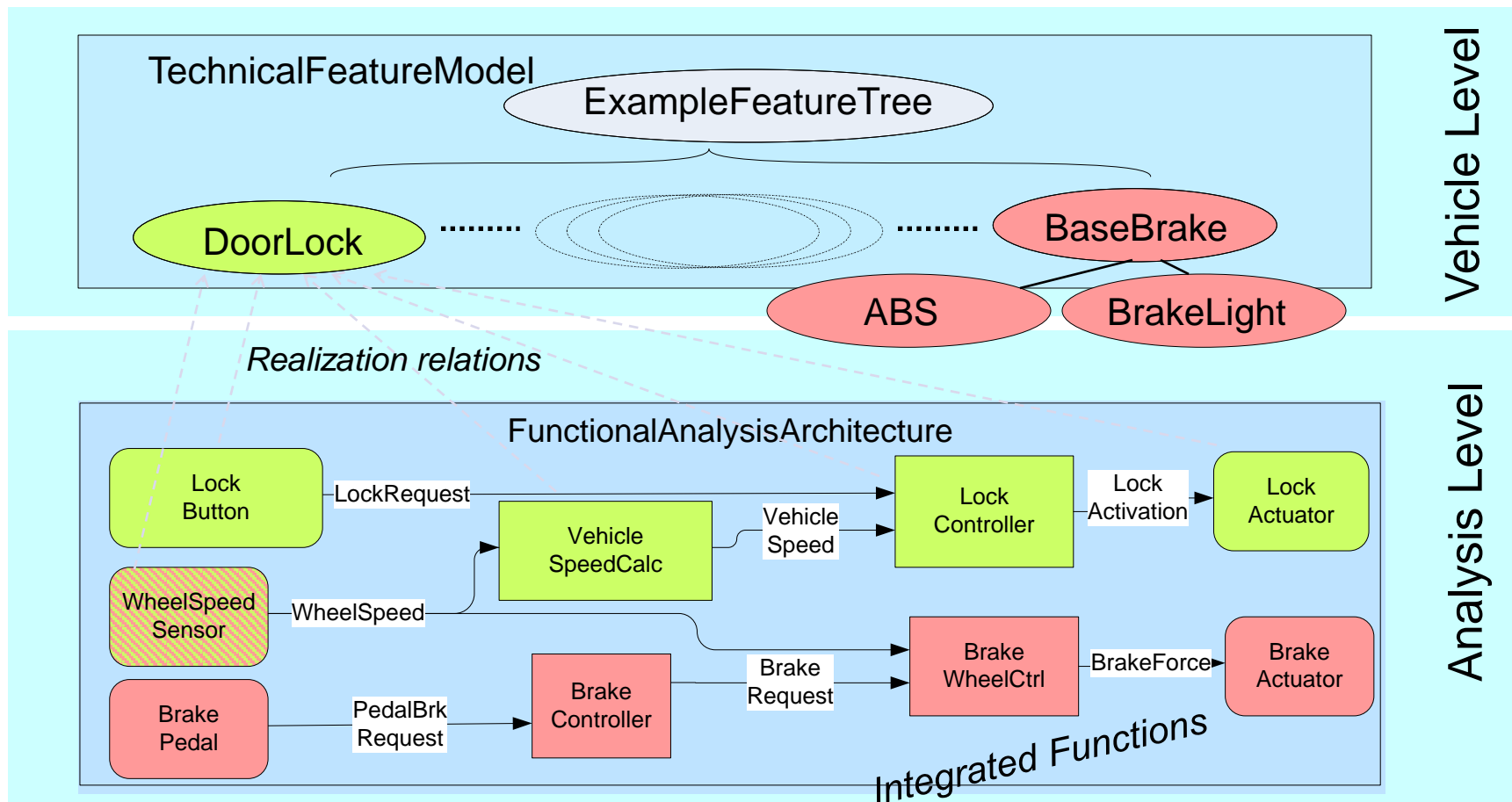
EAST-ADL+AUTOSAR Representation



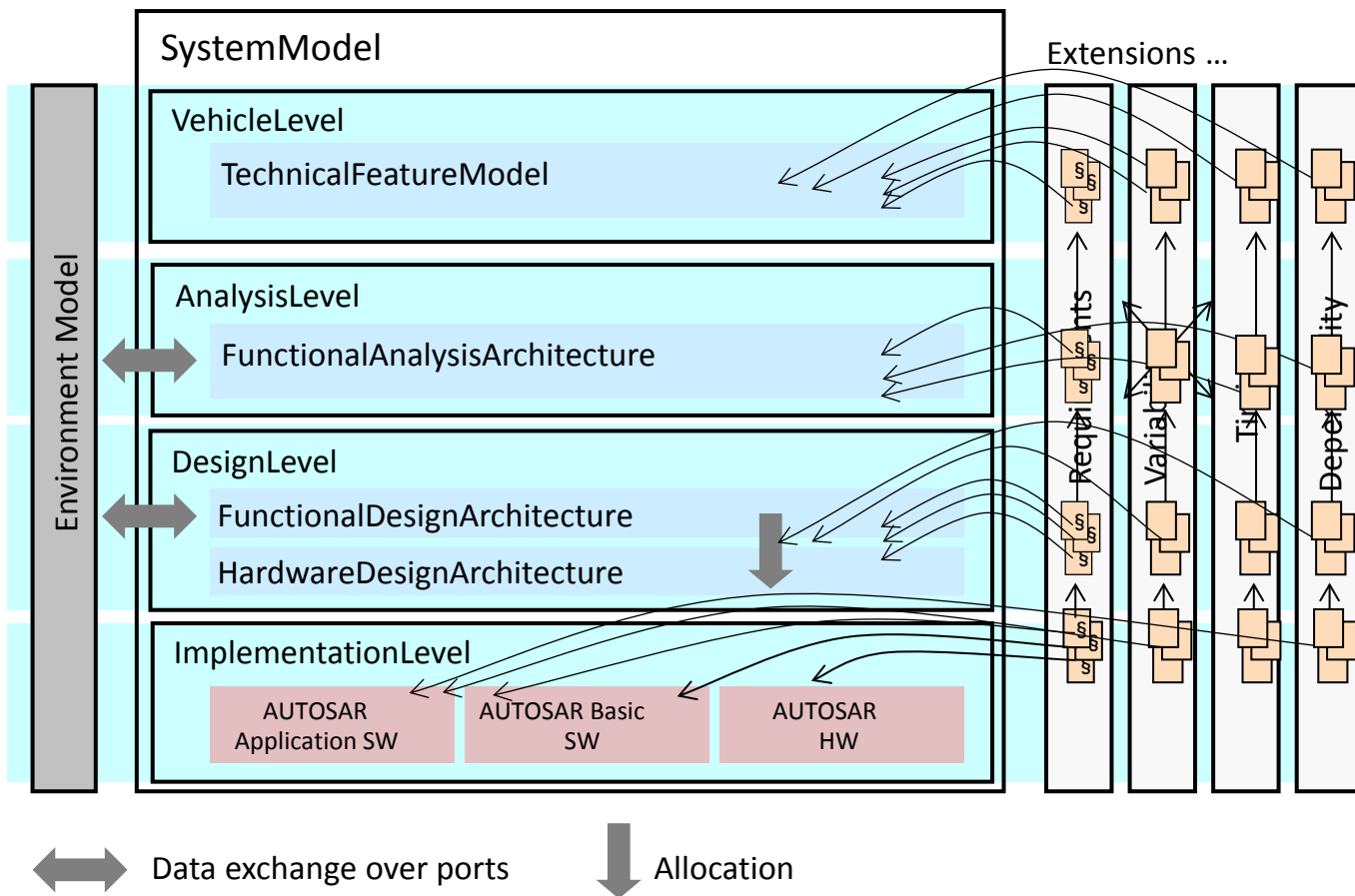
EAST-ADL Abstraction Levels



EAST-ADL Abstraction Levels



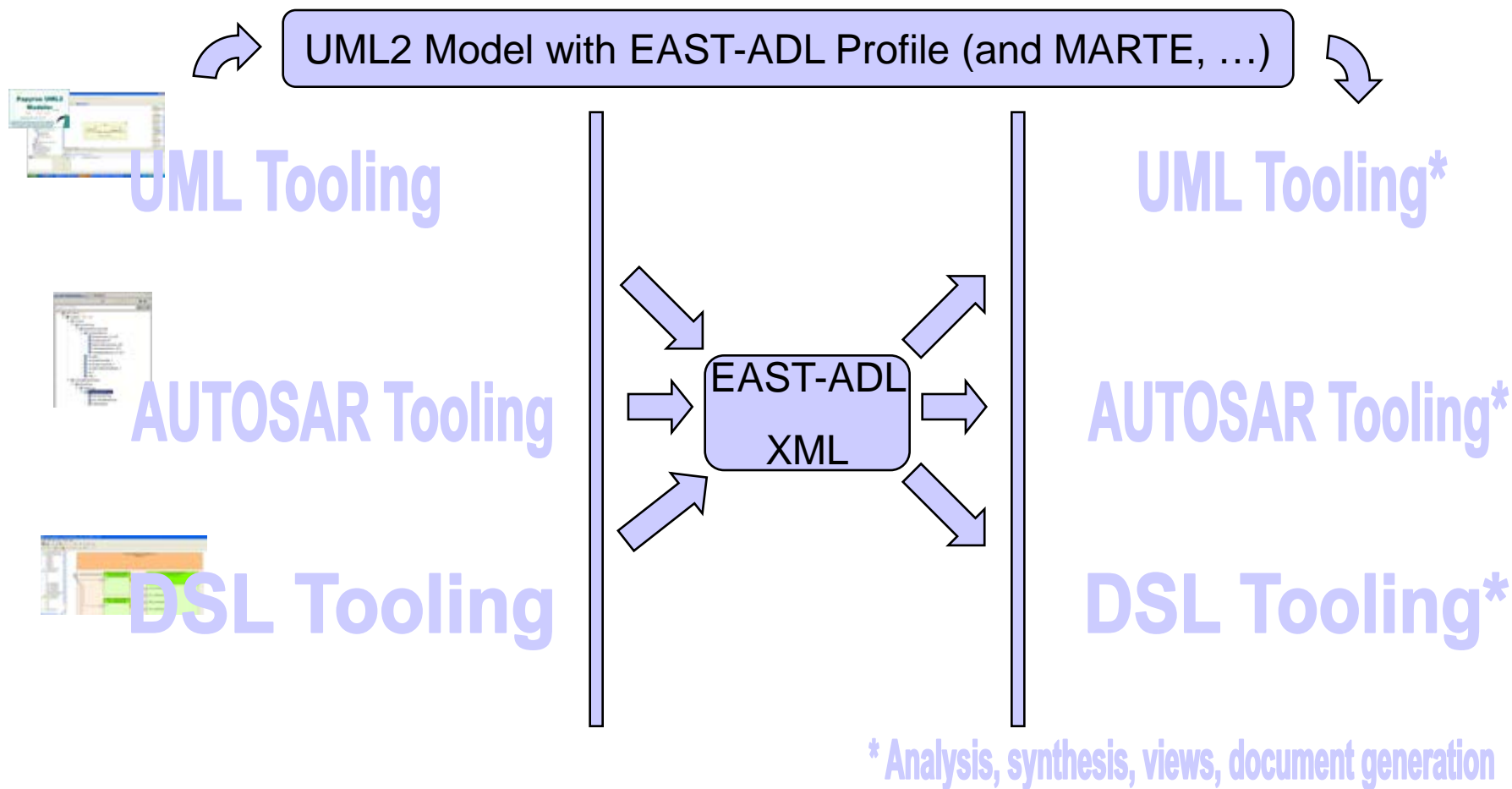
EAST-ADL Extensions



EAST-ADL Extensions: Summary

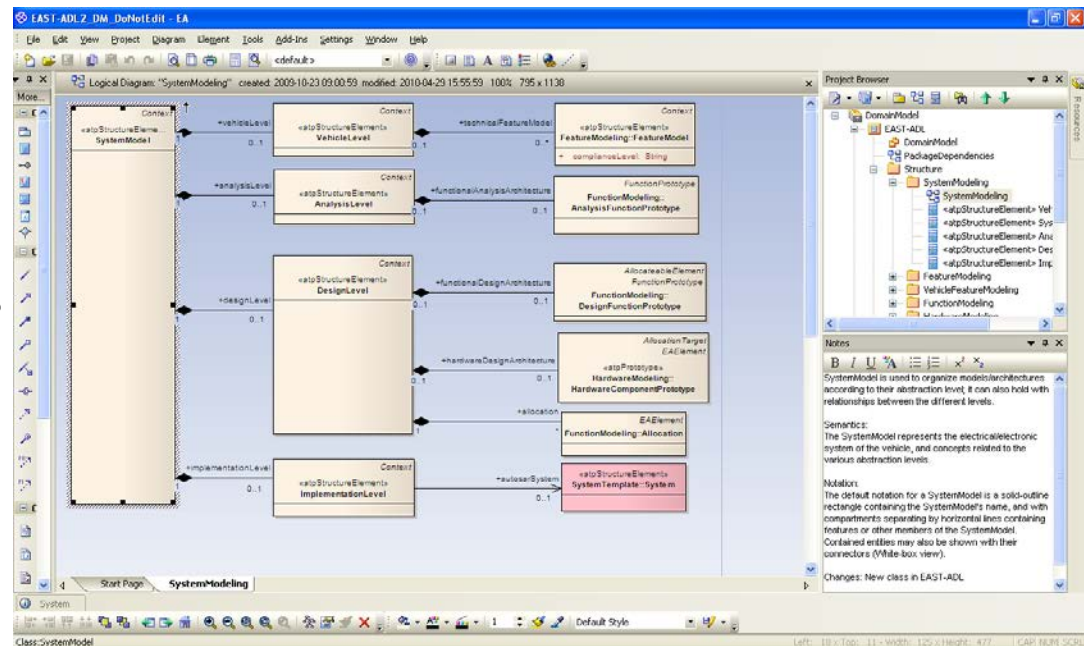
- Constructs for requirements, safety, variability, timing, environment, etc. represents extensions
- Extensions are organized according to abstraction level
- Extensions reference the structural core
- Language may be supported in steps
- UML Profile Application can be modular
- Language annexes can be added
- Changes in one annex does not affect rest
- Extensions can be applied to AUTOSAR

Model Definition and Exchange



Language Definition

- Metamodel defined in Enterprise Architect
- Documentation autogenerated from model
- Exchange format autogenerated using AUTOSAR rules
- AUTOSAR elements can be integrated



Conclusion

EAST-ADL supports automotive embedded systems modelling “starting” with needs and requirements and “ending” with an AUTOSAR SW architecture

- An agreed modelling language makes it possible
 - to understand engineering information from other departments/disciplines and companies
 - to exchange engineering models between different organizations
 - to progress jointly on tools and methodology for modelling, analysis and synthesis

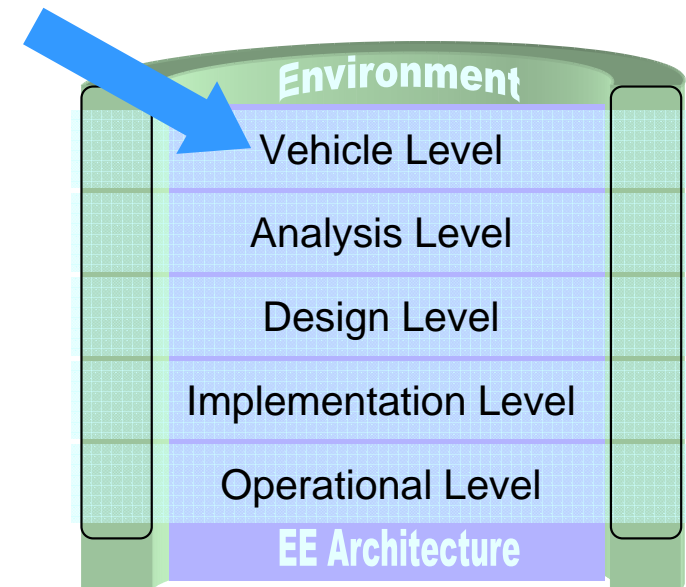
Product Planners decide what to put in the next product

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

Vehicle Feature Model organize Features of
the vehicle

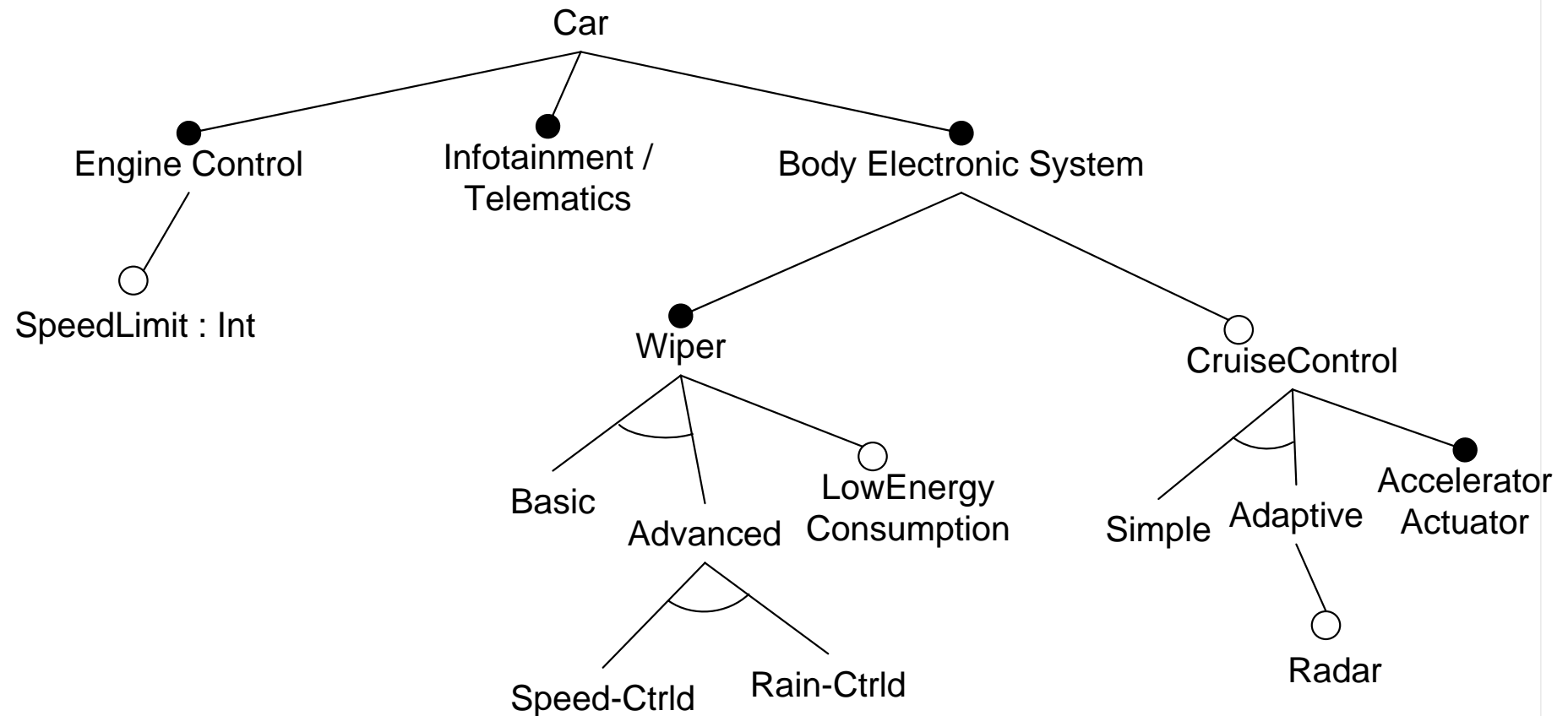
Requirements are included and refer to
external and top level aspects as opposed
to realization details

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



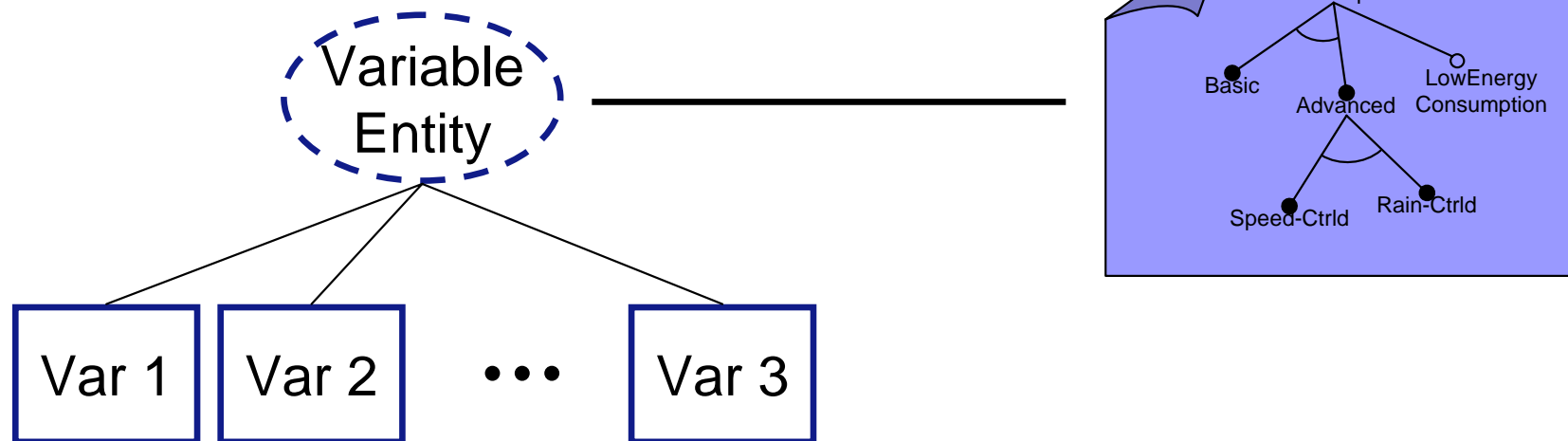
Basic Concepts – Feature Modeling

as introduced by Kang et al. in 1990



What is a Feature ?

A Feature is a characteristic or trait that the variants of a variable entity may or may not have.



What is a Feature ?

A Vehicle is given by a set of Features

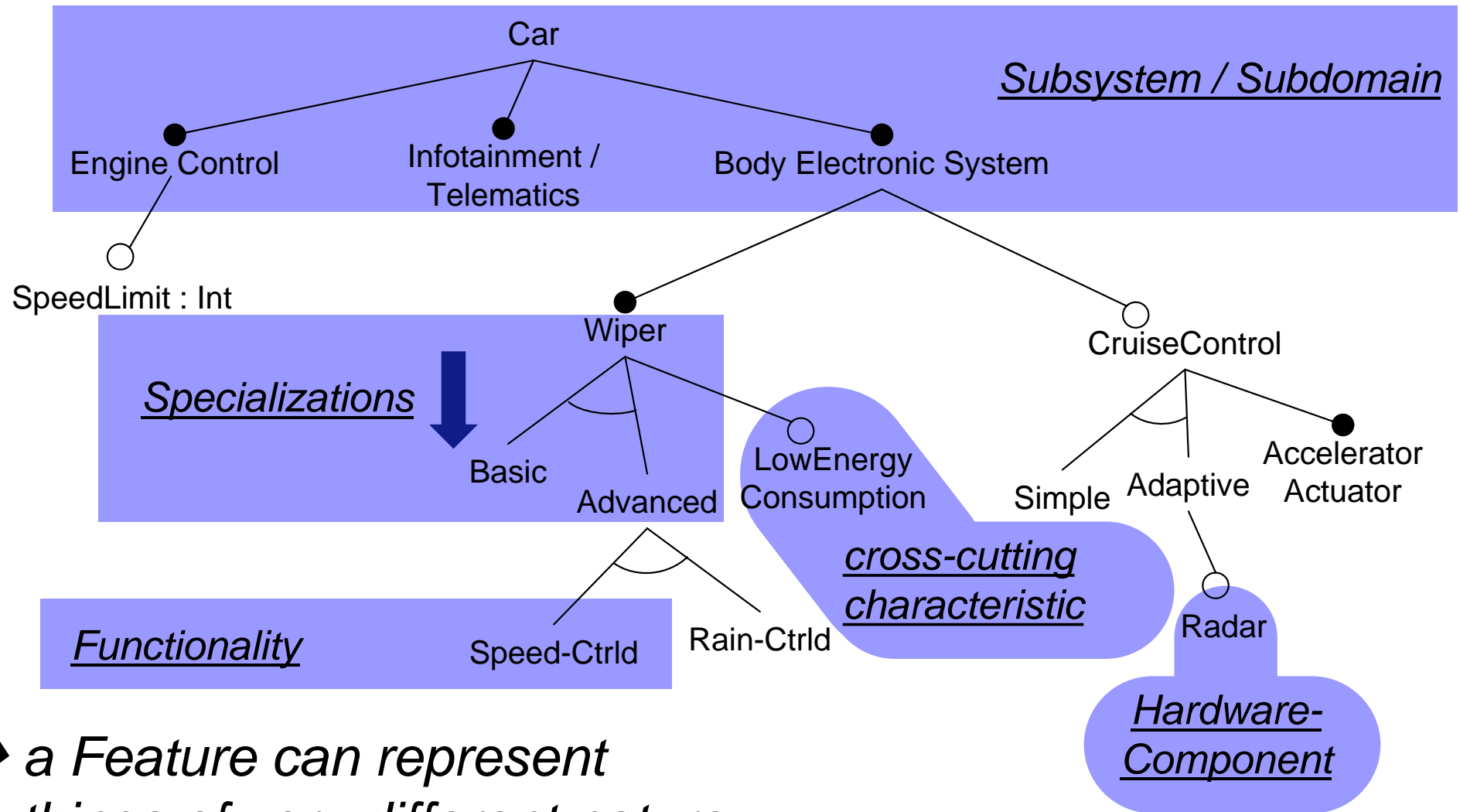
Features are *stakeholder* requested functional or non-functional characteristics of a vehicle

A Feature describes that "what", but shall not fix the "how"

A Feature might be refined by further requirements

From a top-down architecture approach the features are the configuration points to create a vehicle variant

What is a Feature ?



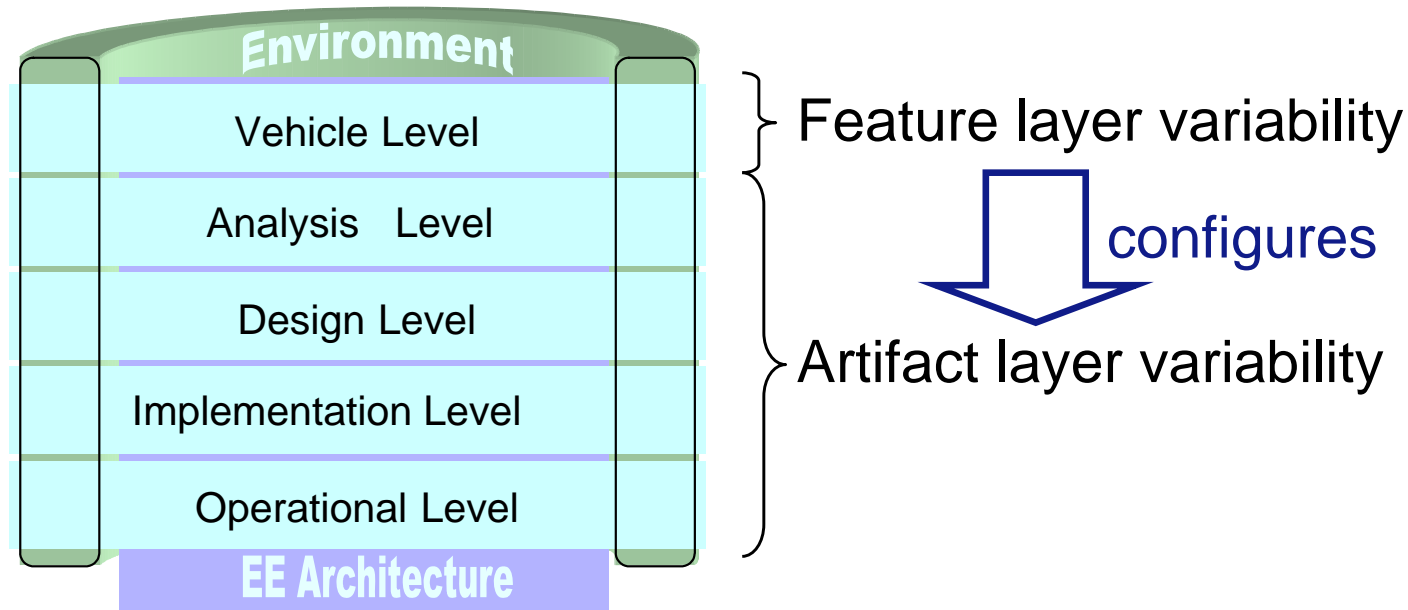
→ a Feature can represent things of very different nature

Variability – Overview

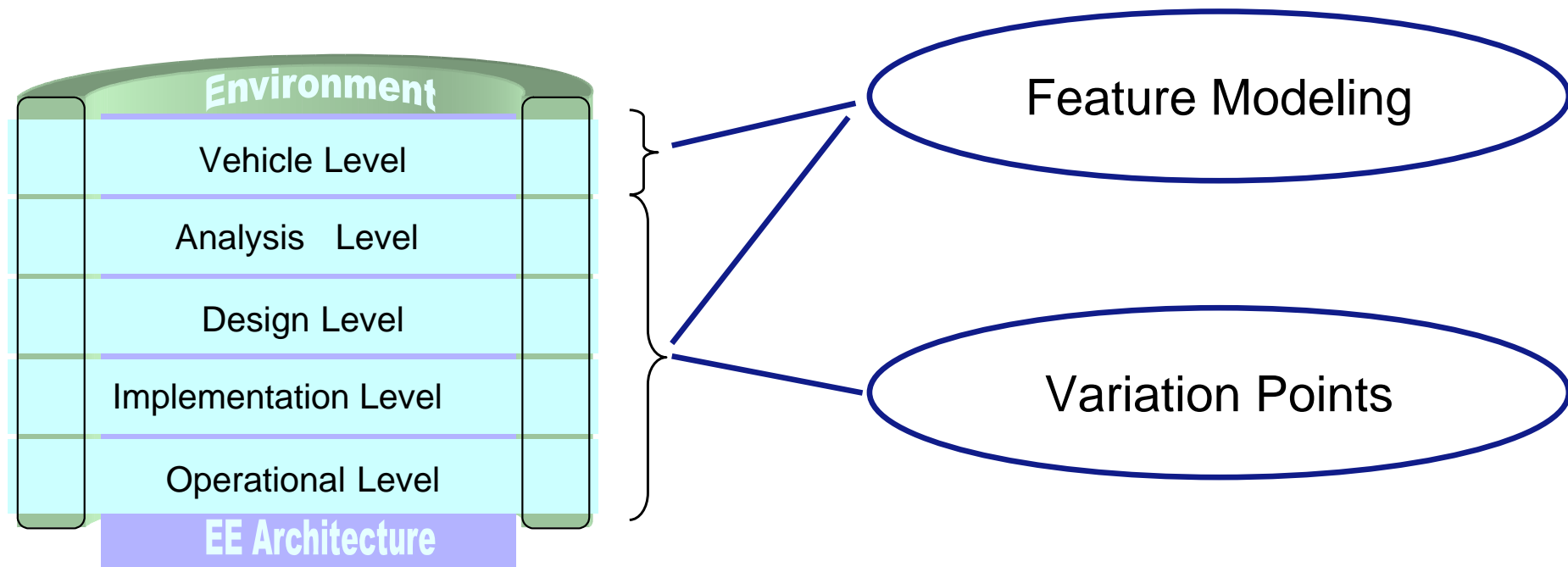
Variability is modeled essentially on two different abstraction layers:

Feature layer variability (being the primary source for variant/product configuration)

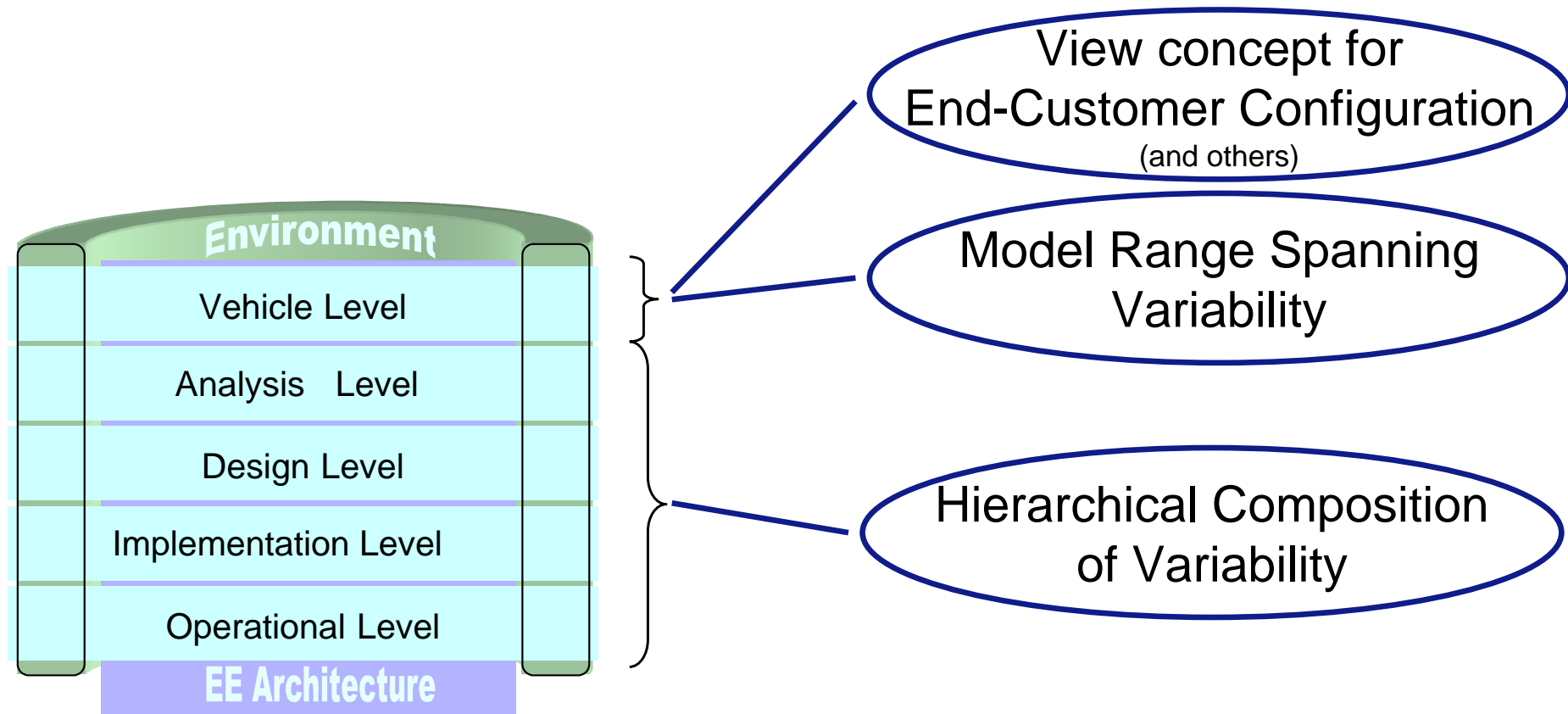
Artifact layer variability (comprising all artifact elements, e.g. requirements, FAA, FDA...)



Basic Concepts



Advanced Concepts



End-Customer Configuration

vehicle level contains the core feature model

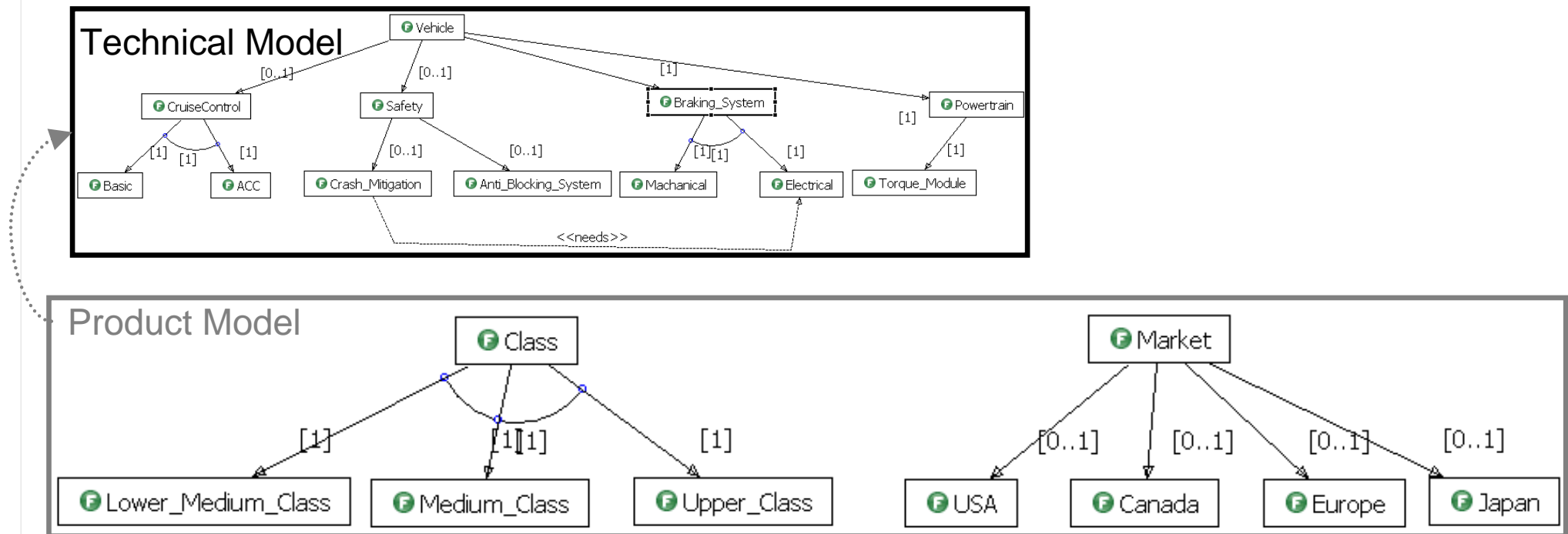
- showing variability of the complete system
- high complexity
- technical viewpoint
(terminology, customer-invisible variability, diverse life-cycle)
- not appropriate for end-customer configuration

vehicle level supports to define end-customer configuration

- provides „view“ on core feature model
- allows for orthogonal „packaging“ of variability
- supports orthogonal configuration considerations
- can be used for separation of other concerns

Vehicle Feature Model and Product Model

A set of product decisions specifies how the Basic Model is configured depending on the configuration of the Product Model:



Example for a product decision: “*All Canadian cars have ACC*”. I.e. if the market “Canada” is selected, in the Basic Model the feature “ACC” is selected.

Multi-Level Feature Model – Example

