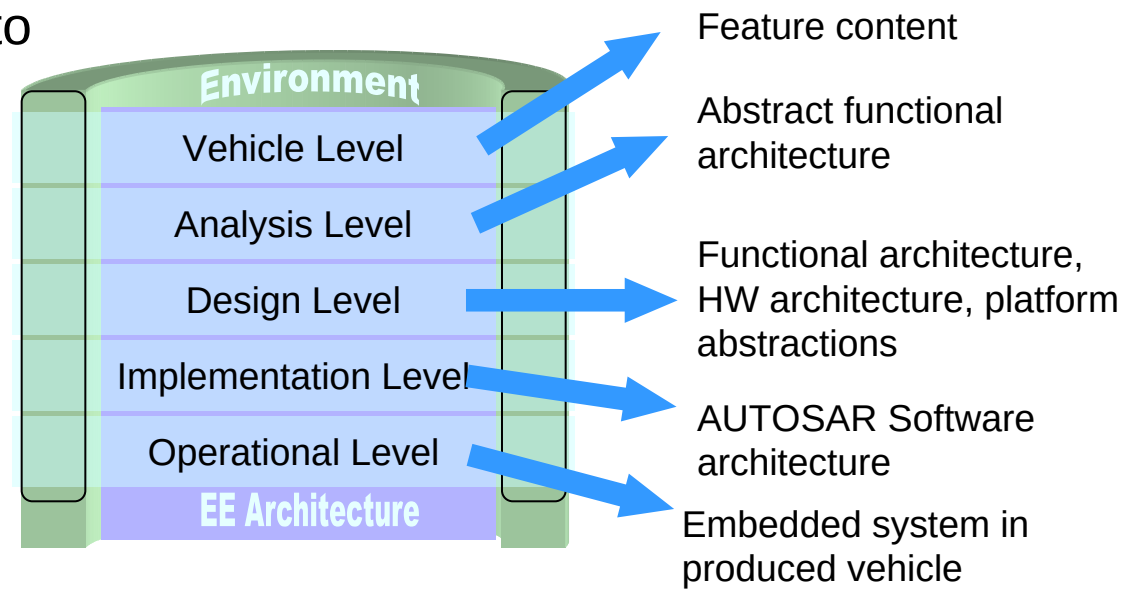
EAST-ADL

A System Modeling Approach 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



Analysis Level

Describes functions realizing the Features

Reflects top level functional decomposition

Allows analysis from a functional/ control engineering point of view

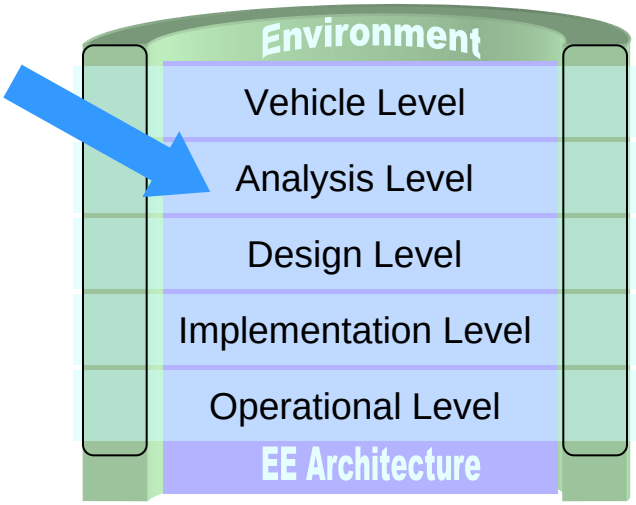
Most implementation details are hidden

Interaction with other functions and environment

Understanding of the function w.r.t algorithmic behavior

Detection of contradicting requirements

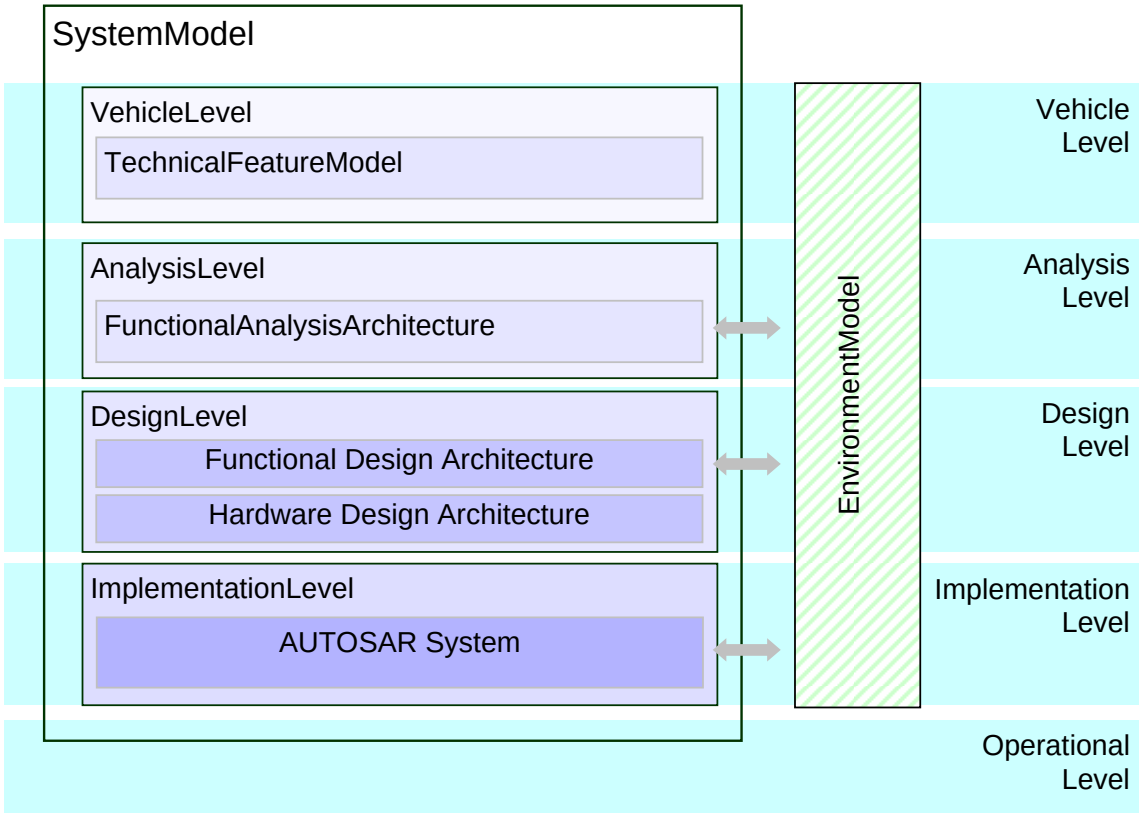
Communication with other departments/stakeholders



Analysis Architecture

Main building blocks:

- AnalysisFunctions
  - Hierarchical decomposition of functionality
- Functional Devices
  - Abstract sensors and actuators



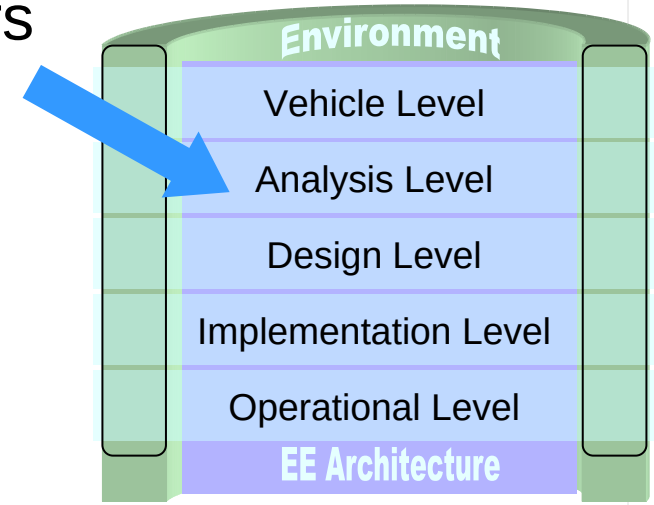
Validation on Analysis Level

Are the requirements consistent?

- Does the system execute at all
- Does it execute “meaningfully”

Identification of critical system parameters

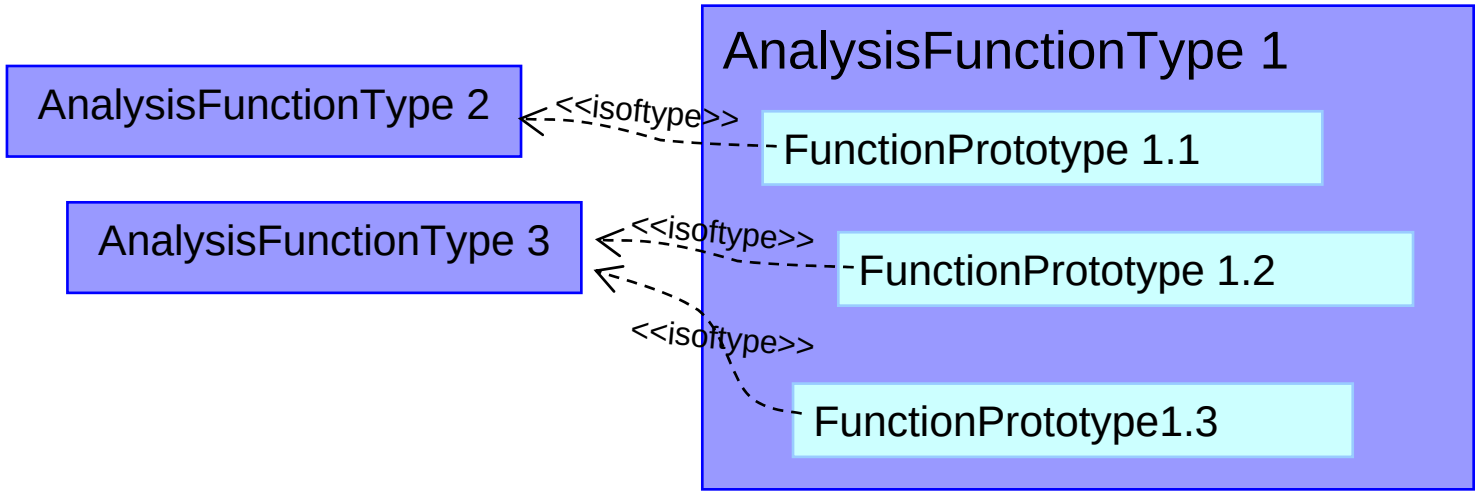
Are the requirements complete?



Function Hierarchy

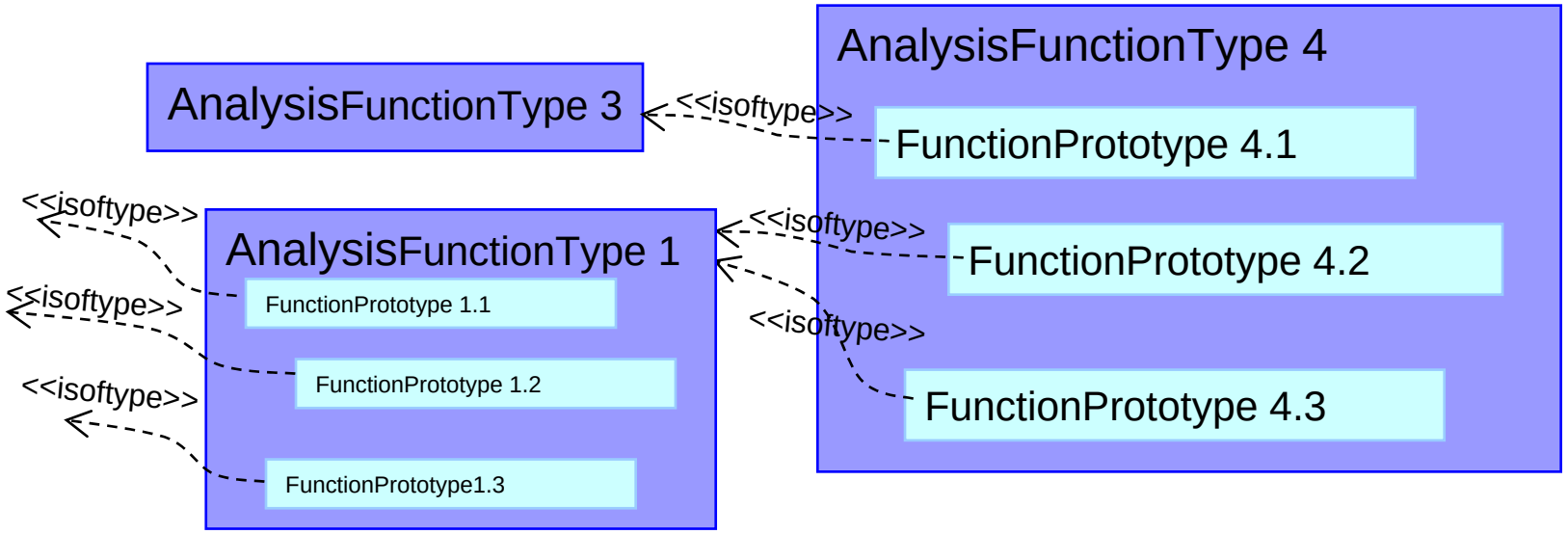
AnalysisFunctions are defined hierarchically using

- Type definitions define the AnalysisFunction and its content
- Prototypes that represent occurences of AnalysisFunctions



Function Hierarchy, contd.

A deep hierarchy can be defined by letting a hierachical Functiontype be the type of a prototype.



EAST-ADL and the Abstraction levels

EAST-ADL models are divided into parts with different abstraction levels

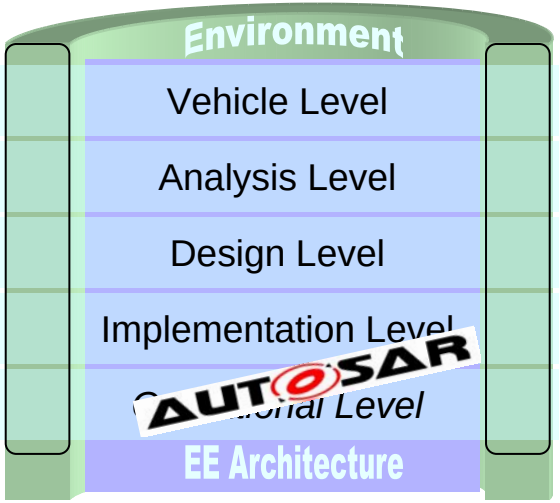
The full EE System is described completely on each abstraction level – “from sensors to actuators”

The Environment is included in the model

The content/concerns and purpose of the abstraction levels differ

Orthogonal aspects:

- **Level of Detail**
- **Level of Integration**
- **Level of Composition**
- **Level of Abstraction**
- **Model of computation**
- **Views**



Design Level - Overview

Models functional decomposition (implementation-driven)

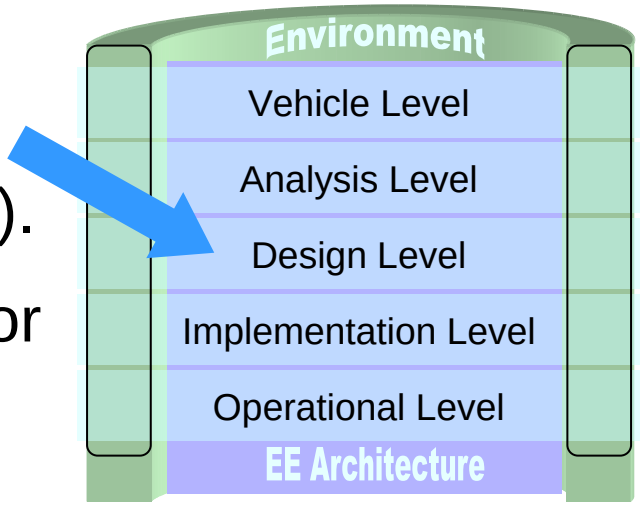
- Functions can be re-allocated w.r.t tasks and ECUs

Signal descriptions using Abstract Data Types

- Parameter - Signal type compliance

Contains implementation level interface aspects (signal data types and timing).

Implementation level sensor and actuator interfacing



Validation on Design Level

Do the Sensor/Actuator Control Algorithms work

Are all control algorithms appropriately implemented

- All required signals available
- All provided signals really used
- Are all reusable signals reused?

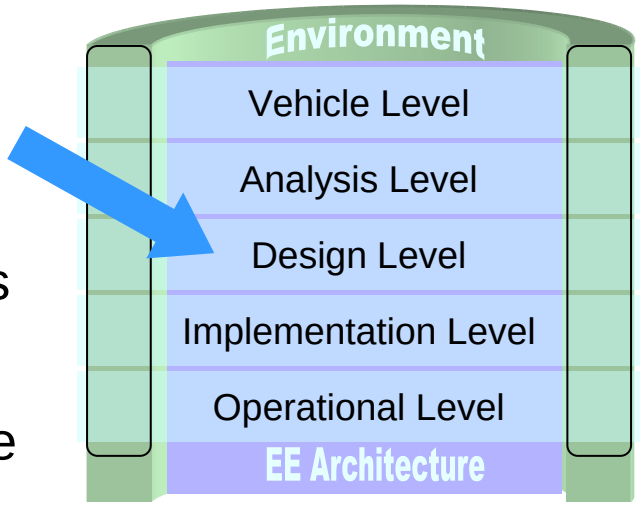
Are the system causal relationships ensured

Are the execution and communication resources adequate

Are there remaining critical single-point-of-failure

Sufficient precision and range of variables

...



Execution Semantics of Functions

Elementary DesignFunction

- Synchronous Behavior:
  - **Run-to completion**
  - **Read All Inputs before execution**
  - **Compute**
  - **Write to Outputs**
- Time or event trigged

Composite DesignFunction

- Contains Connectors and prototypes of DesignFunctions
- Precedence constraint to manage execution order
- Governing DesignFunction to manage execution order

From Sensor to Actuator

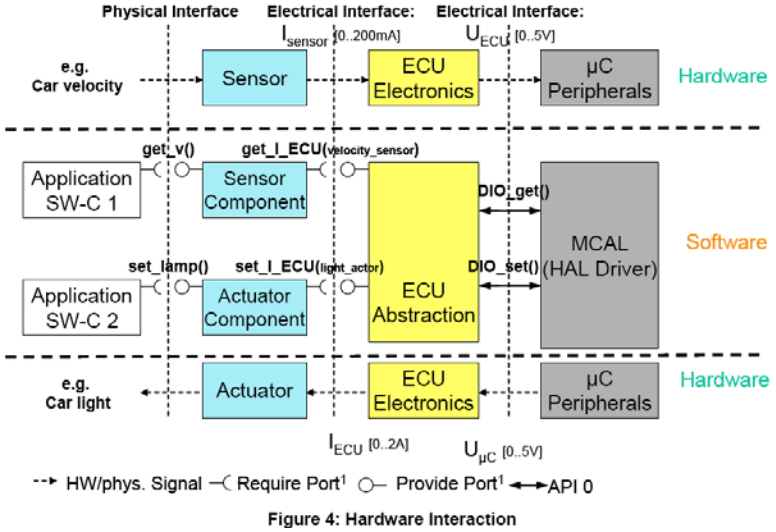
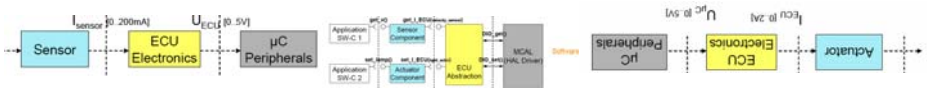
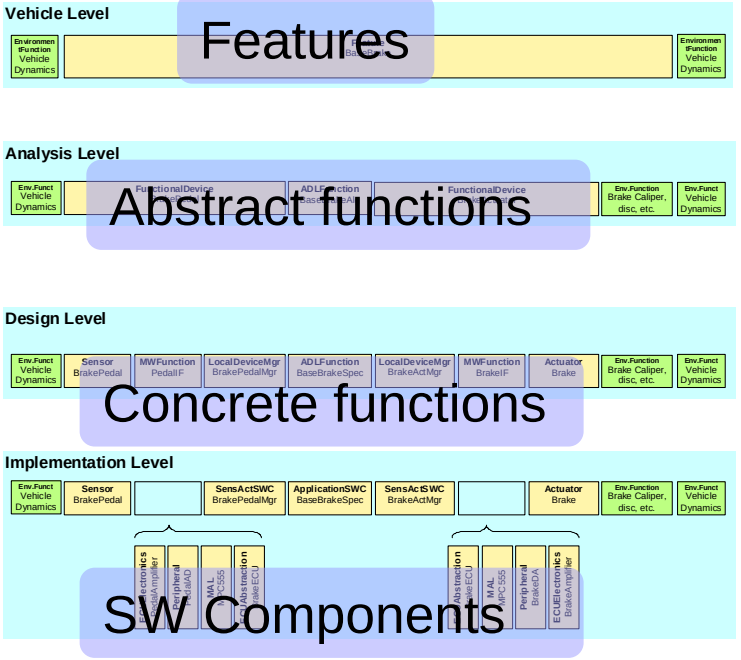
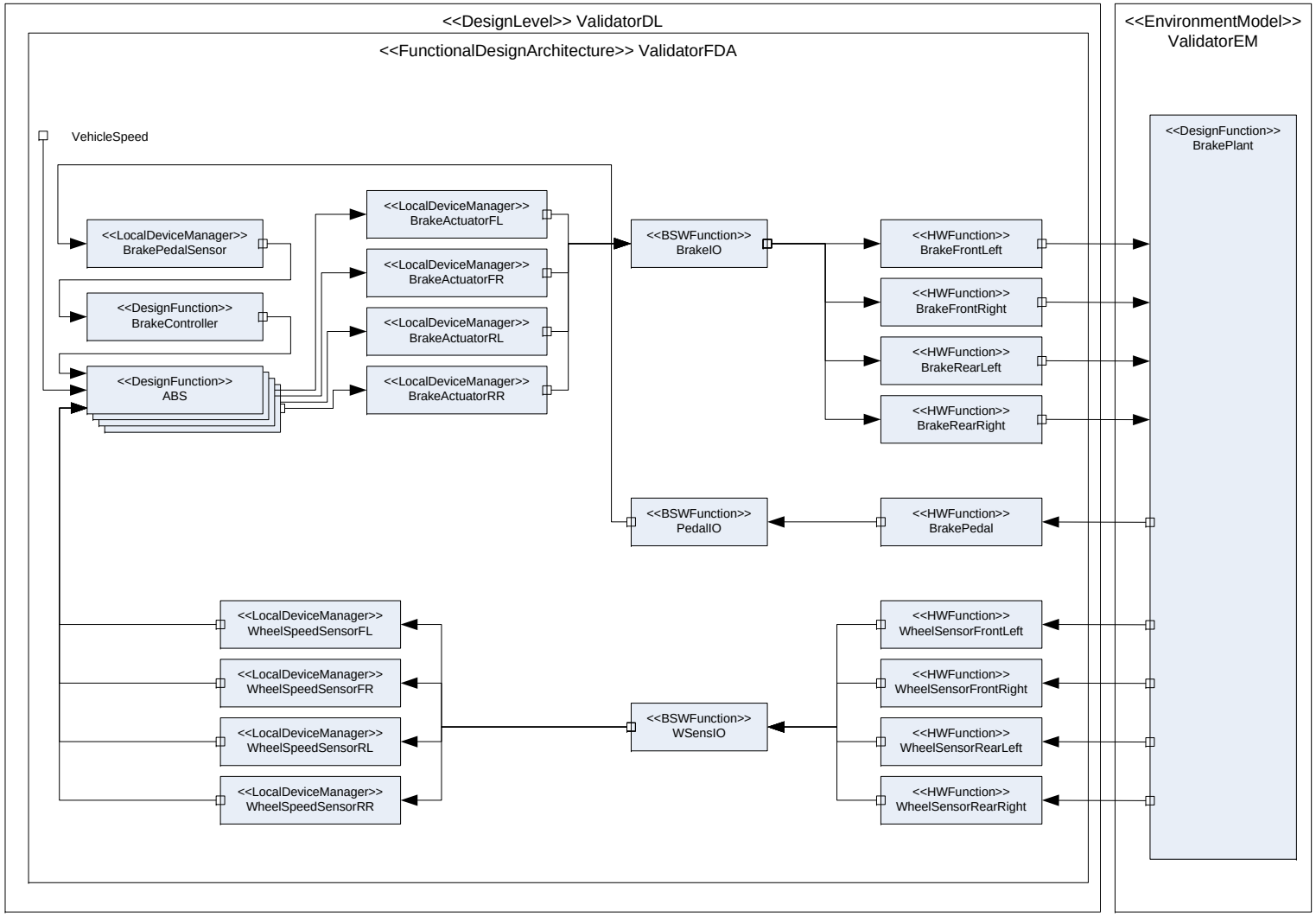


Figure 4: Hardware Interaction

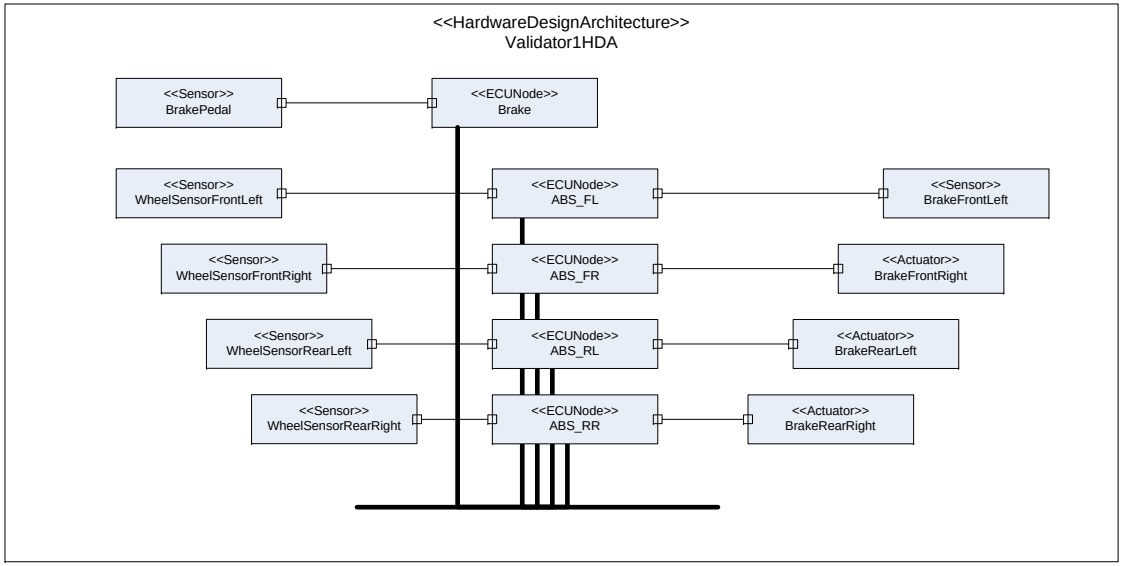




Example – Functional architecture



Example – Hardware Architecture



EAST-ADL Behavior

Some Purposes of a Behavioral definition:

- Behavioural Specification
  A definition of intended behavior for documentation purposes
- Behavioural Simulation
  A definition of intended behavior for simulation
- Behavioural Analysis
  A definition of intended behavior for (formal) analysis of properties
- Behavioural Synthesis
  A definition of intended behavior for configuration, code generation, etc.

EAST-ADL Behavior

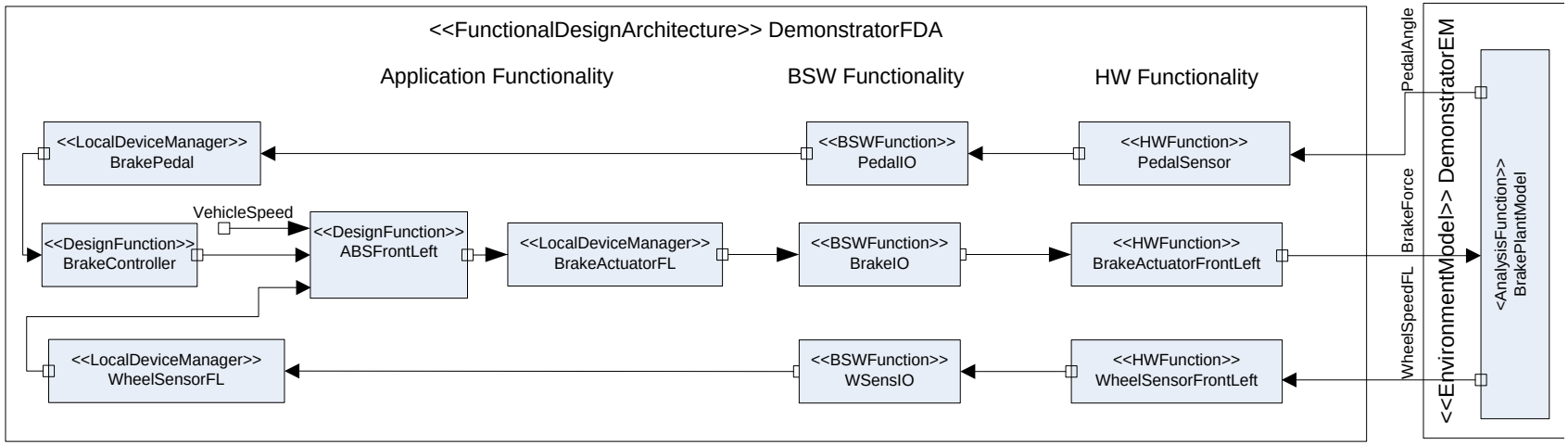
EAST-ADL is primarily a structural representation

Behavior aspects includes:

- Execution semantics
- Transfer functions
- Integrated Behavior of systems/subsystems

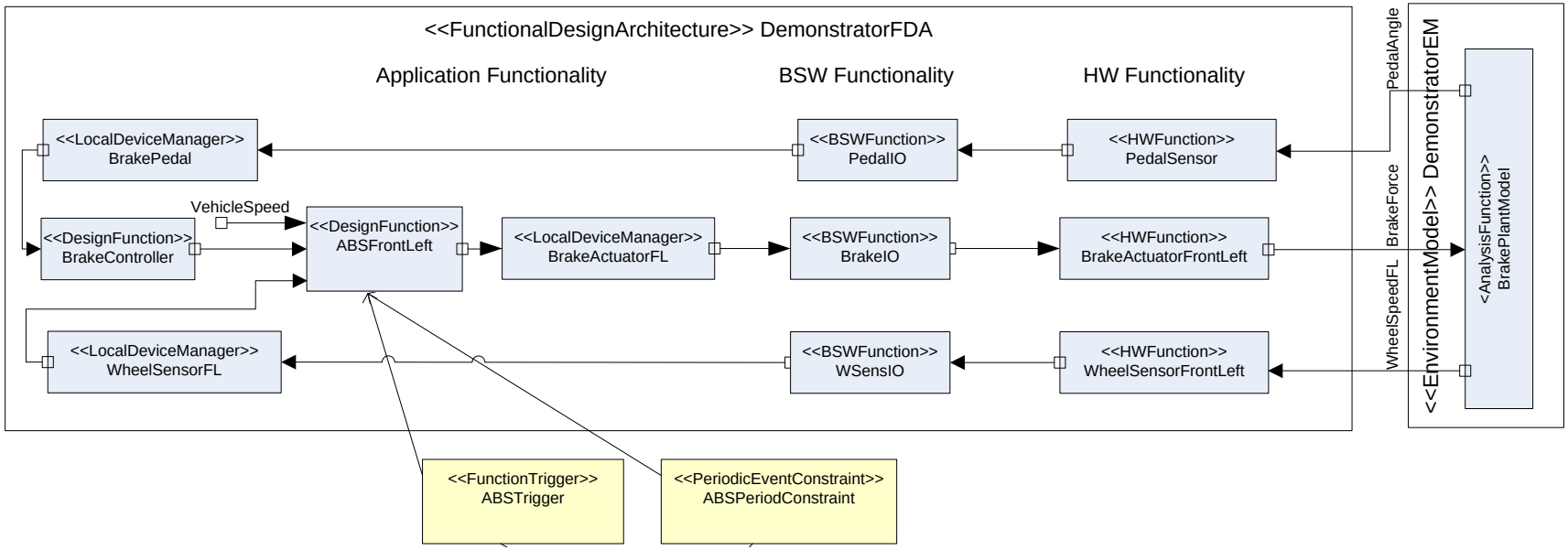
EAST-ADL Model on Design Level

Structure



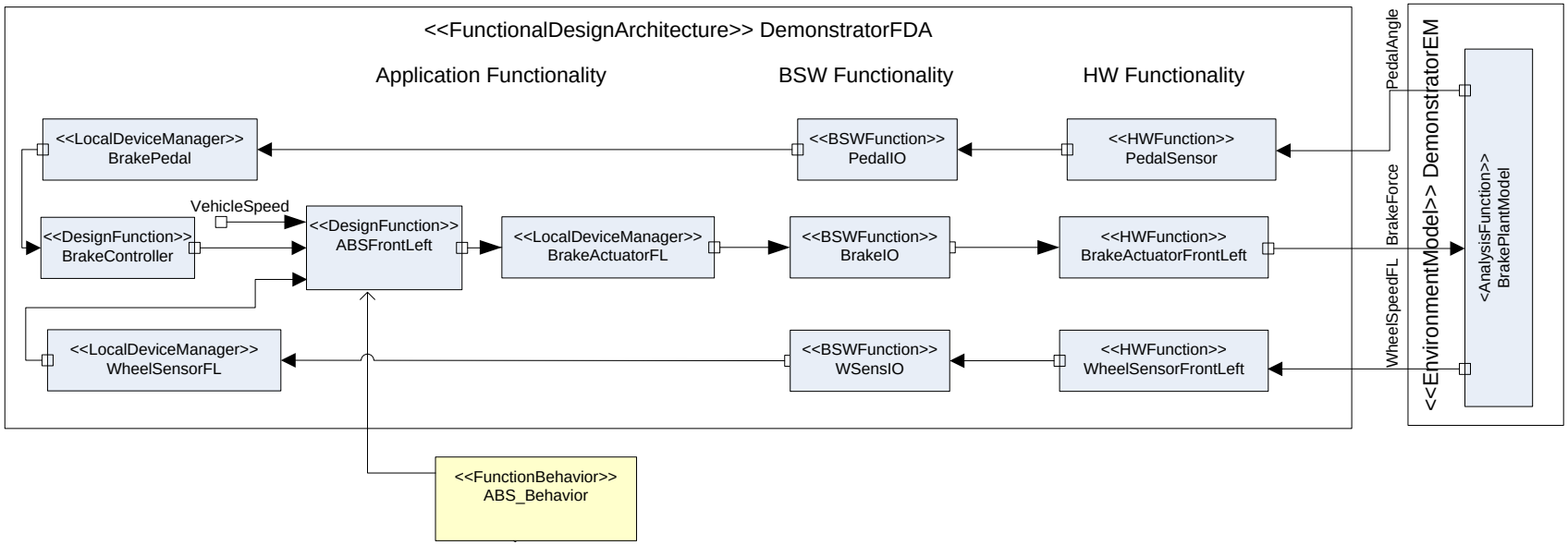
EAST-ADL Model on Design Level

- Timing/Triggering



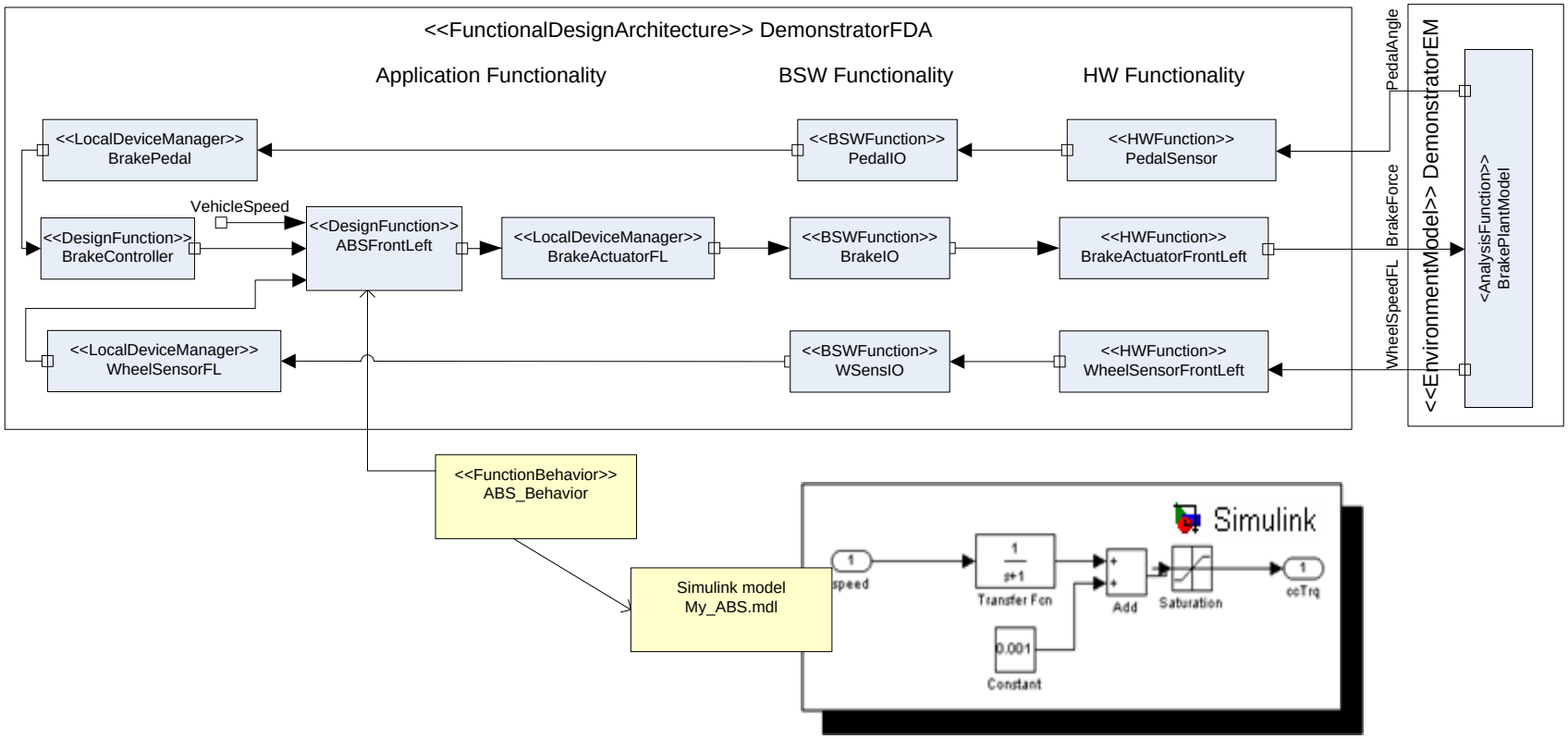
EAST-ADL Model on Design Level

Transfer Function – ”Black-box” behavior



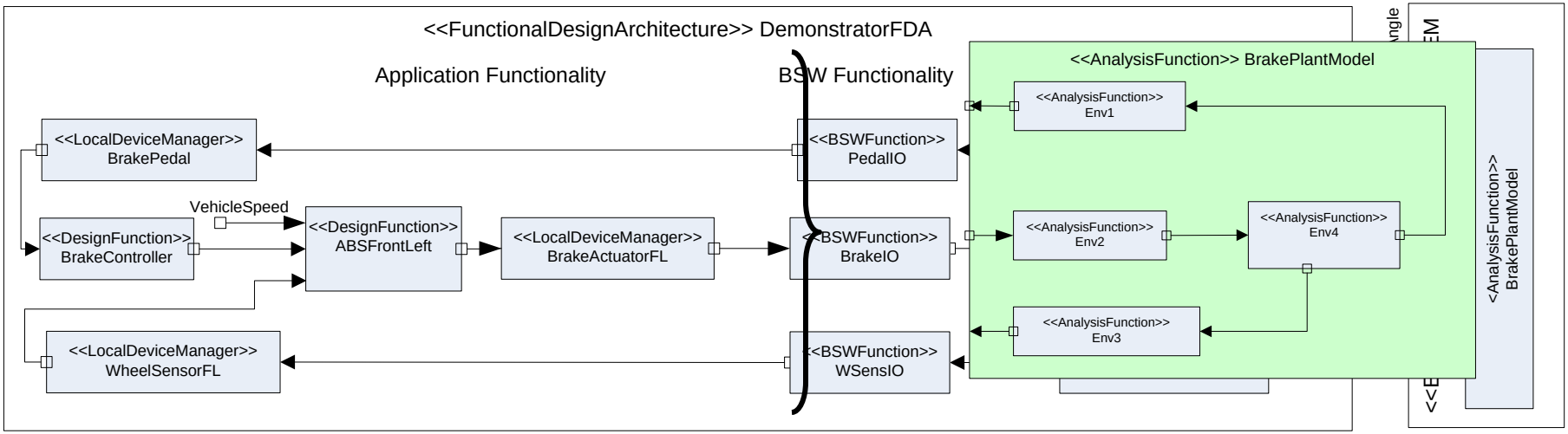
EAST-ADL Model on Design Level

Transfer Function – ”Black-box” behavior



EAST-ADL Model on Design Level

Behavior of environment (Plant)

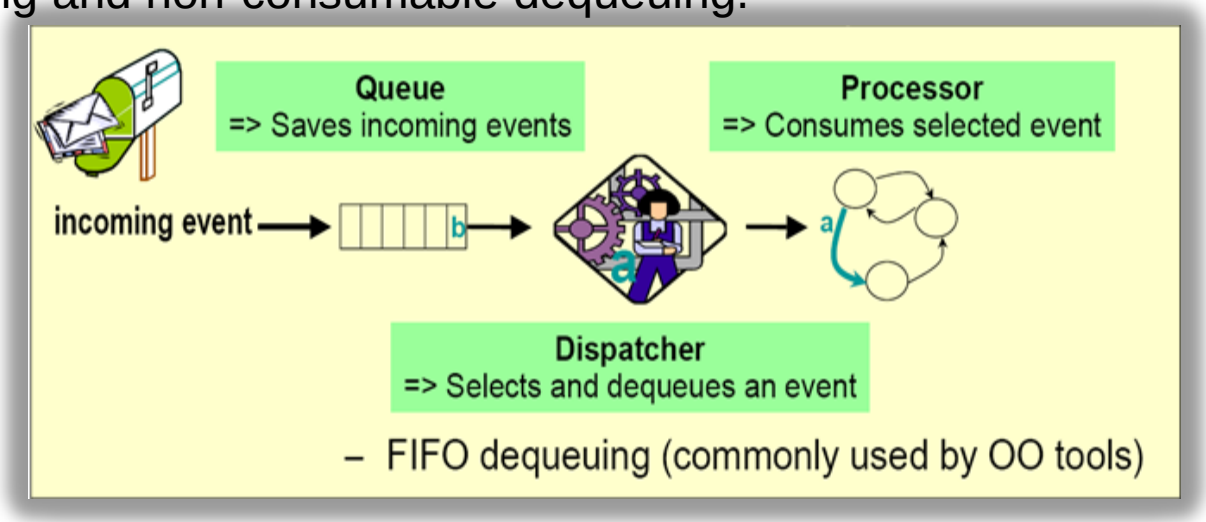


Typically in plant model:

- Non-causal: ”Power Ports” – {Torque,Speed}, {Pressure, Flow}, etc.
- Continuous Time: No triggers

Semantics of Function Behavior

- Computation(transfer function): follows the semantics of the external representation used (e.g. Simulink or StateMate).
- Execution: synchronous
  1. Read inputs from input ports
  2. Execute Behavior with fixed inputs (run-to-completion)
  3. Provide outputs to output ports
- The targeted ports are single size buffers with non-blocking access, overwritable queuing and non-consumable dequeuing.



Behavioral notations

Commercial tools

- Simulink
- Ascet
- Scade
- StateCharts
- ...

Open Notations

- UML State charts, Activity diagrams
- SPIN
- UPPAAL
- Modelica
-

Model Structure must be respected

- Inputs, outputs

Execution Semantics must be respected

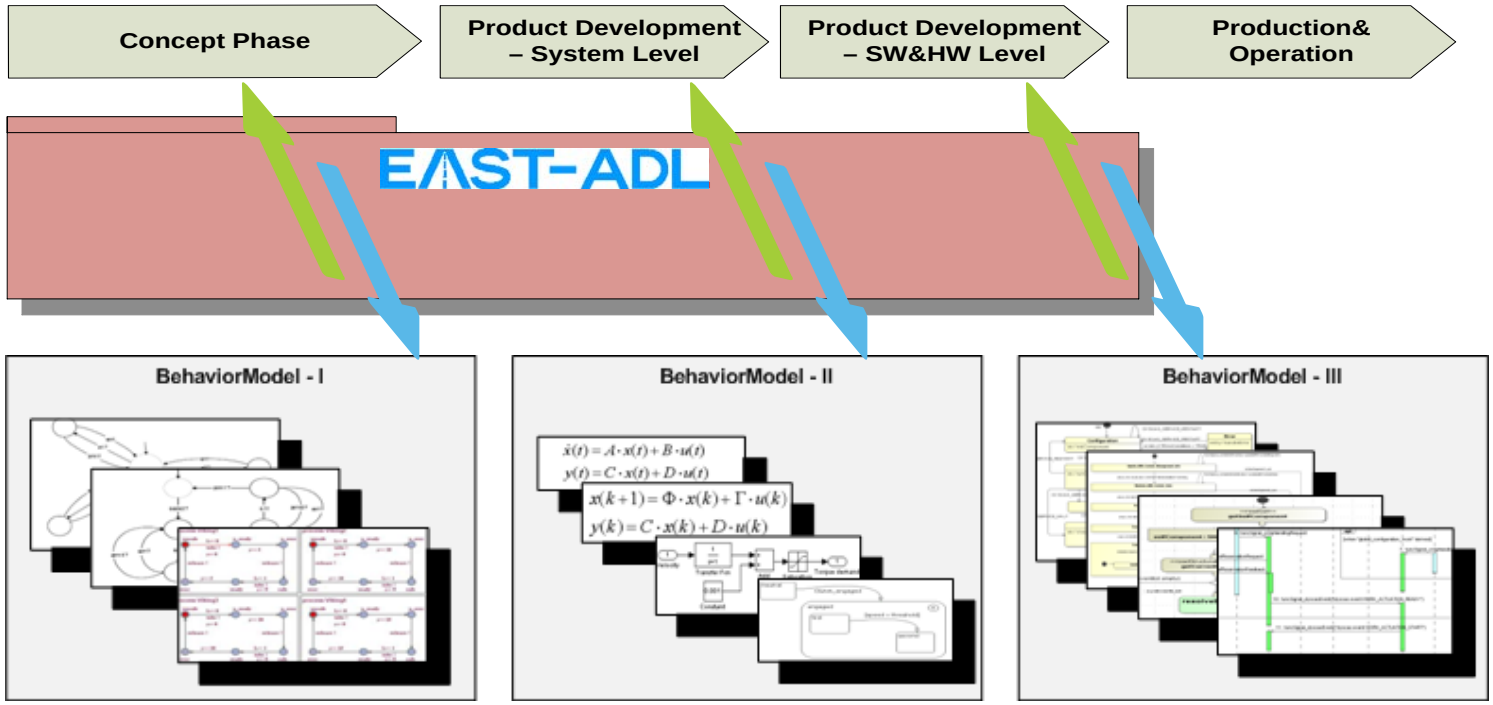
- Continuous/discrete
- Triggering
- Data Exchange

Support for Modes

- Modes
  - Declarative modes – ”assuming mode X, this is what happens”
  - Transfer functions, constraints, triggers refer to mode(s)
- Mode Groups
  - Mutually exclusive set of modes: {driver_present, driver_absent}, {parked, stand-still, in-motion}
  - The realization of mode switches and mode notifications is part of ”black-box” behavior

Generic Roles of Behavior Modeling

Defining system modes, computation/transfer functions, execution dynamics, and related error behaviors.
Refining textual requirement, operation situation and use case statements.
Supporting communication&comprehension, analysis and V&V (e.g., test case generation).



EAST-ADL Behavior Modeling

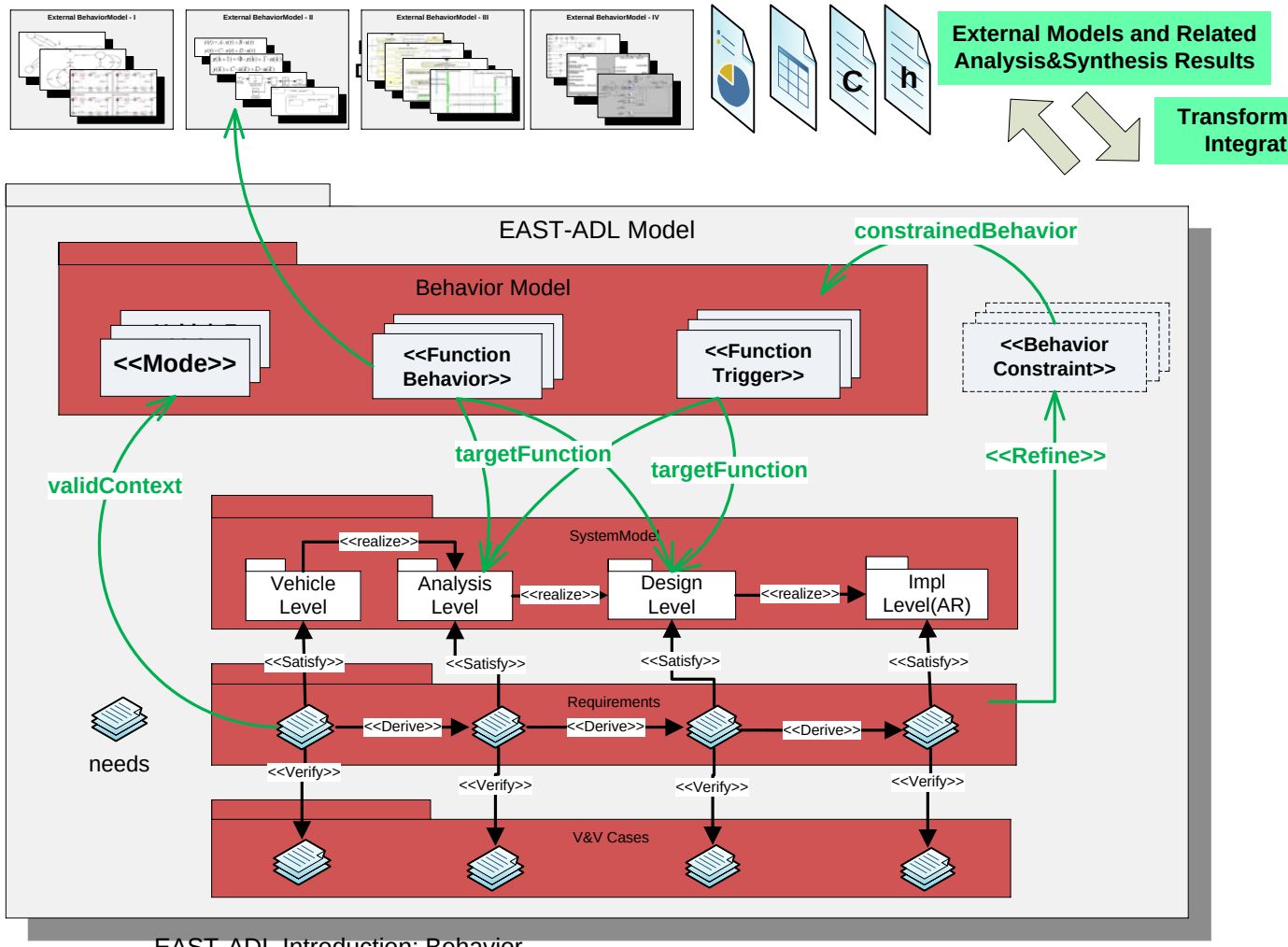
System Behavior

System Structure

Requirement

V&V Definition

Lifecycle Stages



Concept Phase | Product Development – System Level | Product Development – SW&HW Level

Summary

- EAST-ADL Provides Structure and Triggering
  - Leaf Functions execute synchronously (EE-system) or non-causally (Plant/Environment)
- ”Transfer Function” is defined in external tools and notations

Purpose:

- Allow integration of models from different sources
- Allow simulation, analysis, synthesis of integrated models

Work is ongoing to also define a native behavior for ”behavioral constraints”

Summary, cont'd

Purposes of Behavior definition

- Supporting System definition:
  modes, function behavior, function trigger
- Supporting Error definition
  Error behaviors
- Formalizing textual descriptions in requirements and related statements.
- Facilitating analysis and V&V (e.g., test case generation).
- Dedicated support for physical interaction:
  Power ports.
- "input language" to external models and formalisms (e.g., Simulink, SPIN)