

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

EAST-ADL and AUTOSAR

Explaining how EAST-ADL complements the
Software Architecture definition of AUTOSAR

ATESST2 Concept presentation 2010 Q2



General

EAST-ADL:

- to support the engineering effort for automotive embedded systems

AUTOSAR

- to capture the software architecture*

*Architecture in the sense of components and their relationships to each other

2 Kinds of Differences

- **Abstraction Levels:**

- EAST-ADL complements AUTOSAR with higher levels of abstractions
 - Vehicle Level
 - Analysis Level
 - Design Level

- **Engineering Information Scope:**

- EAST-ADL complements AUTOSAR with more concepts
 - Requirements Engineering
 - Variant Management
 - Timing
 - Safety
 - Behaviour (nominal/error)

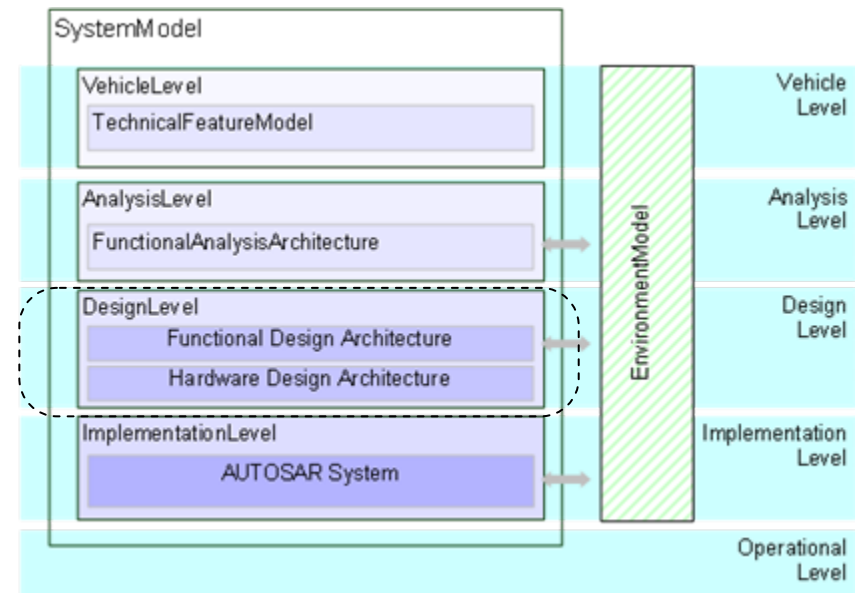
Scope in
AUTOSAR
depending
on version

The Abstraction Level Difference EAST-ADL Defines

(on Design level)

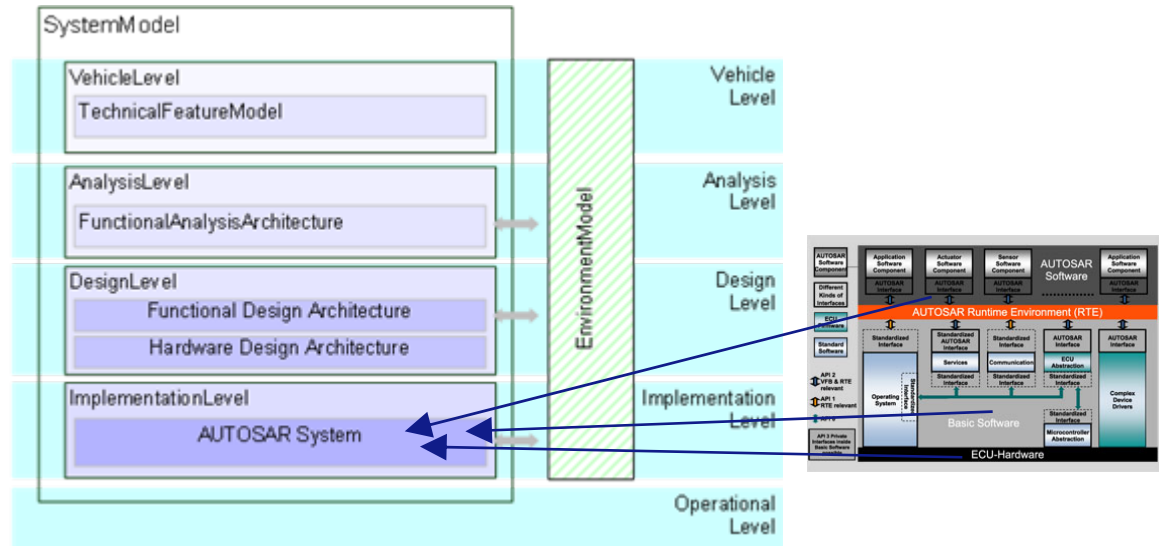
- Hardware entities/topology
- Functional structure & behavior
- Function-to-ECU allocation

These engineering decisions are
constraints for AUTOSAR SW
Architecture and mapping



AUTOSAR defines

- Hardware entities and topology with enough detail to support SW configuration
- Software components with runnables
- Mapping to tasks and frames
- Mapping to ECUs and busses



Software vs. Functional Architecture

Software and functional architecture is orthogonal

Software architecture

- A system decomposition from an implementation viewpoint corresponding to the final product

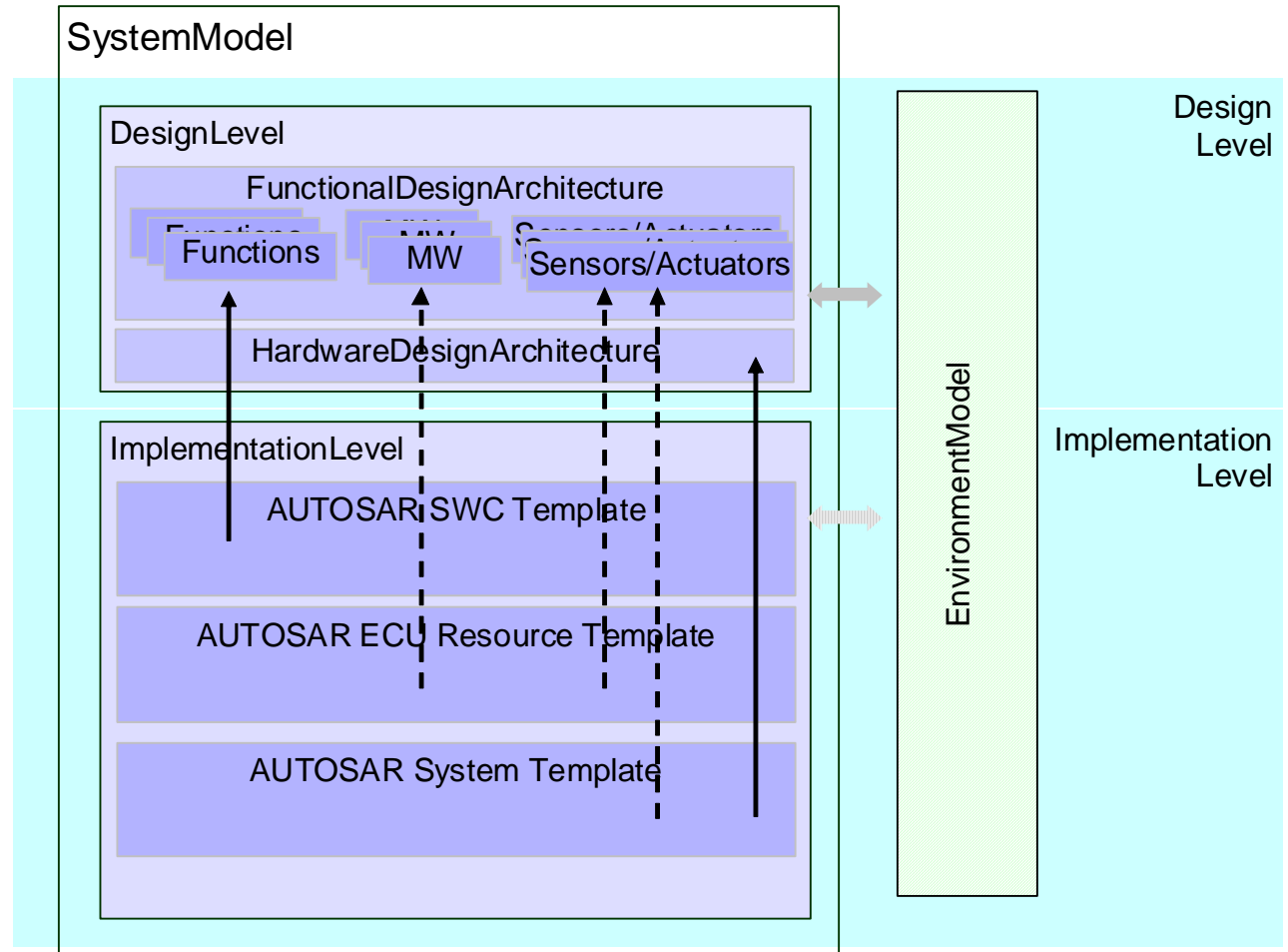
Functional architecture

- A system decomposition from an functional viewpoint defining the logical parts of the system and how they interact

The same functional architecture may be “packaged” in several ways resulting in different software architectures

EAST-ADL concepts capture information that is the rationale for an implementation description using AUTOSAR concepts

Structural Compliance



Mapping of EAST-ADL vs. AUTOSAR

Runnable is the behavioural entity in AUTOSAR
(SWC is structure)

Function represents structure and behavior in EAST-ADL

=> Fundamental mapping is 'Function – Runnable'

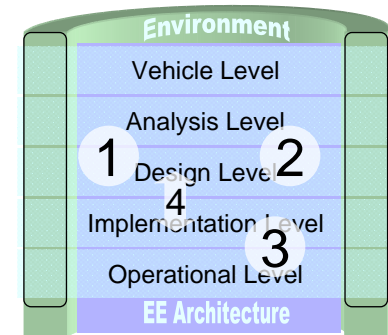
Different mappings are possible

- 'Function – AR Composition'
- 'Function – AR SW Component'

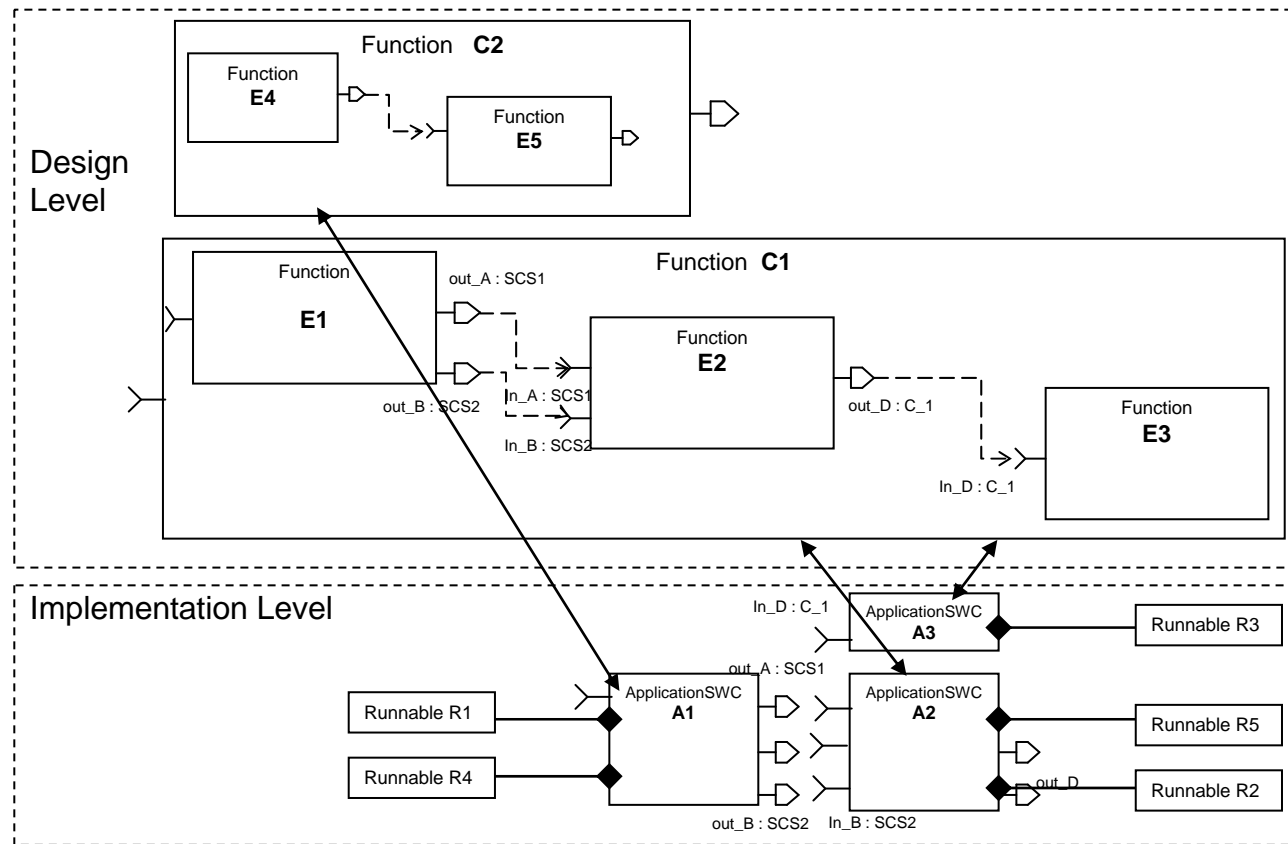
('1 Function – n Composition/SW Component/Runnable' is incorrect)

A possible methodology

1. Define functional structure in EAST-ADL
2. Optional: Identify blocks that should go together in a SW component and put constraints regarding components
3. Define AR SWC and Runnables
4. Map elementary or composite Functions to appropriate AR SWC or Runnable (Realization relation)
 - the behavior of the runnable is defined in the corresponding Function
 - the packaging into SWC/runnables is independent of functional structure
 - SW architecture can be traced back to functions, features, requirements

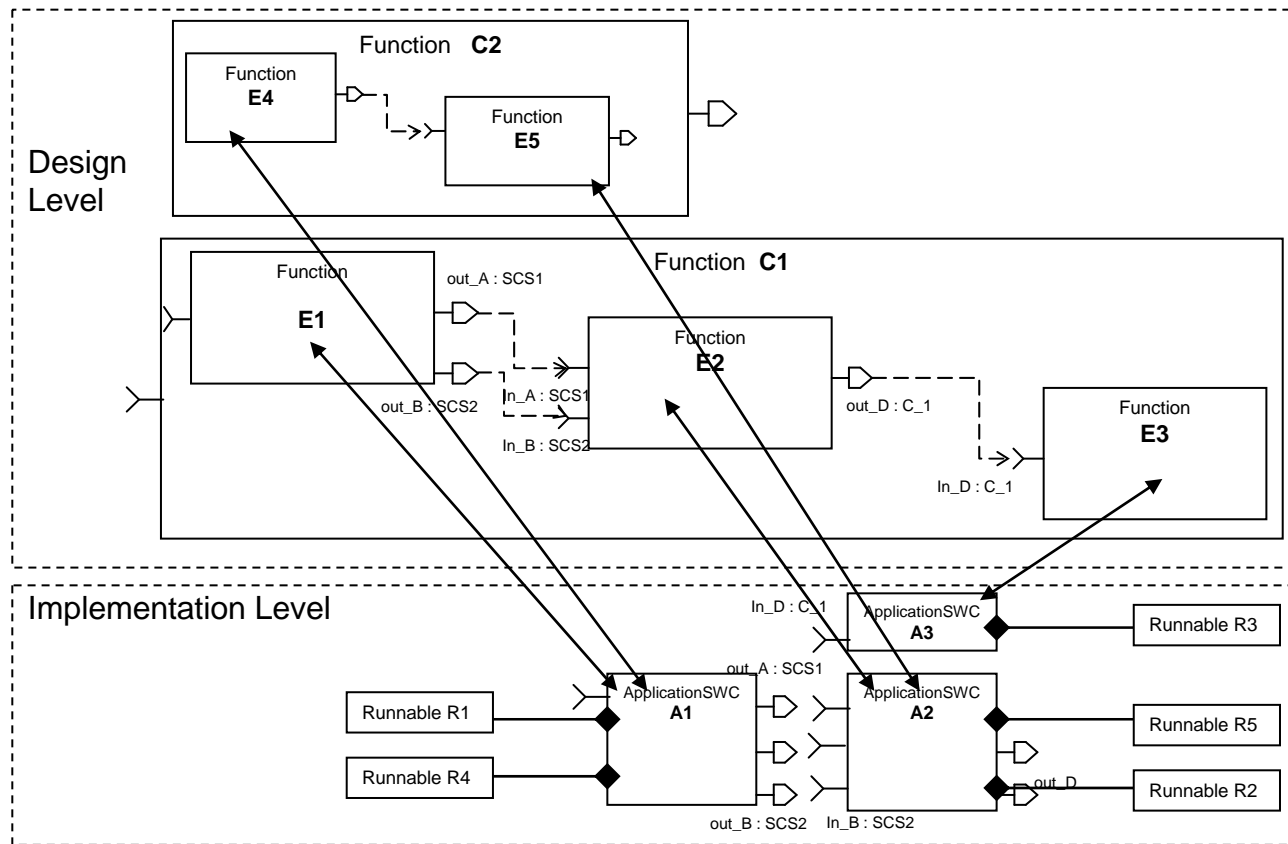


Examples of Function-to-component Mappings



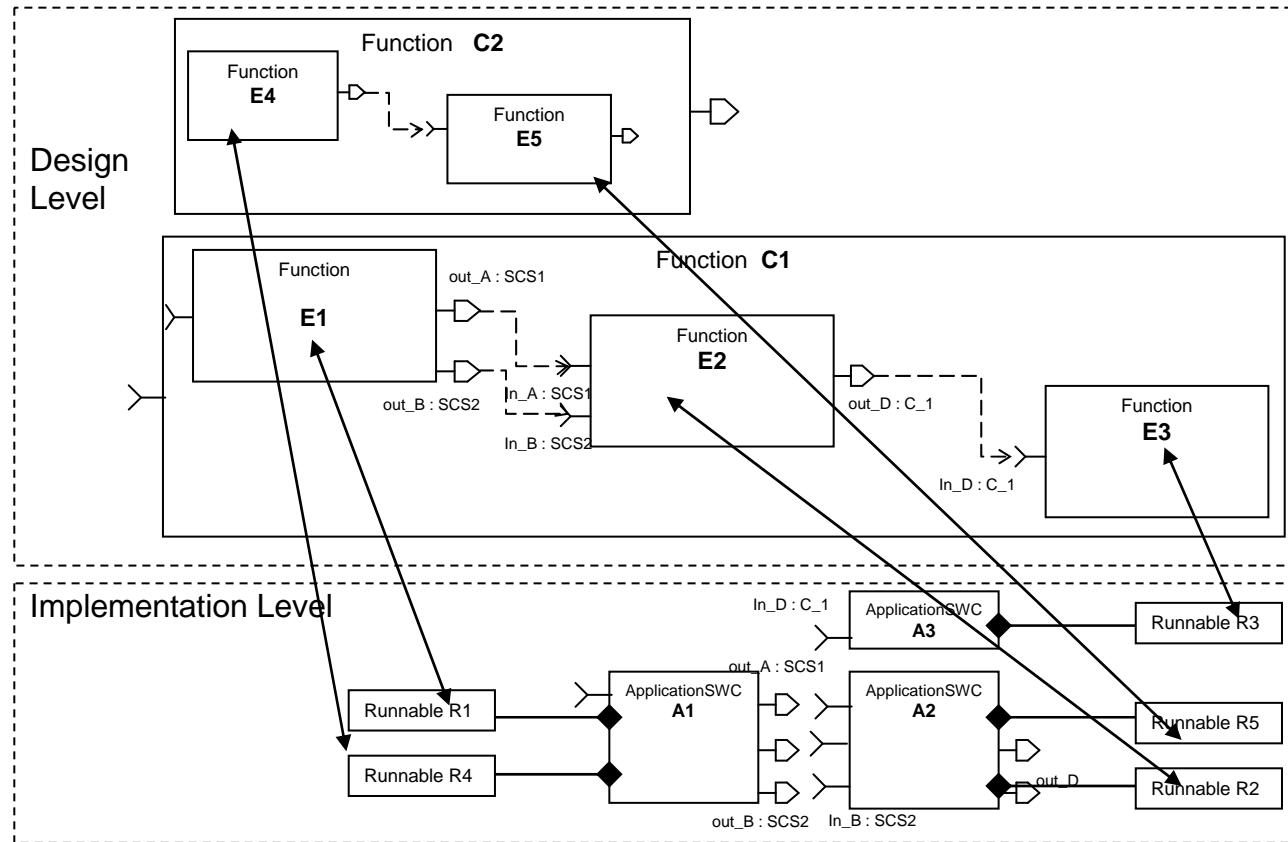
Function to SW Component

Examples of function-to-component Mappings



n Function to 1 SW Component

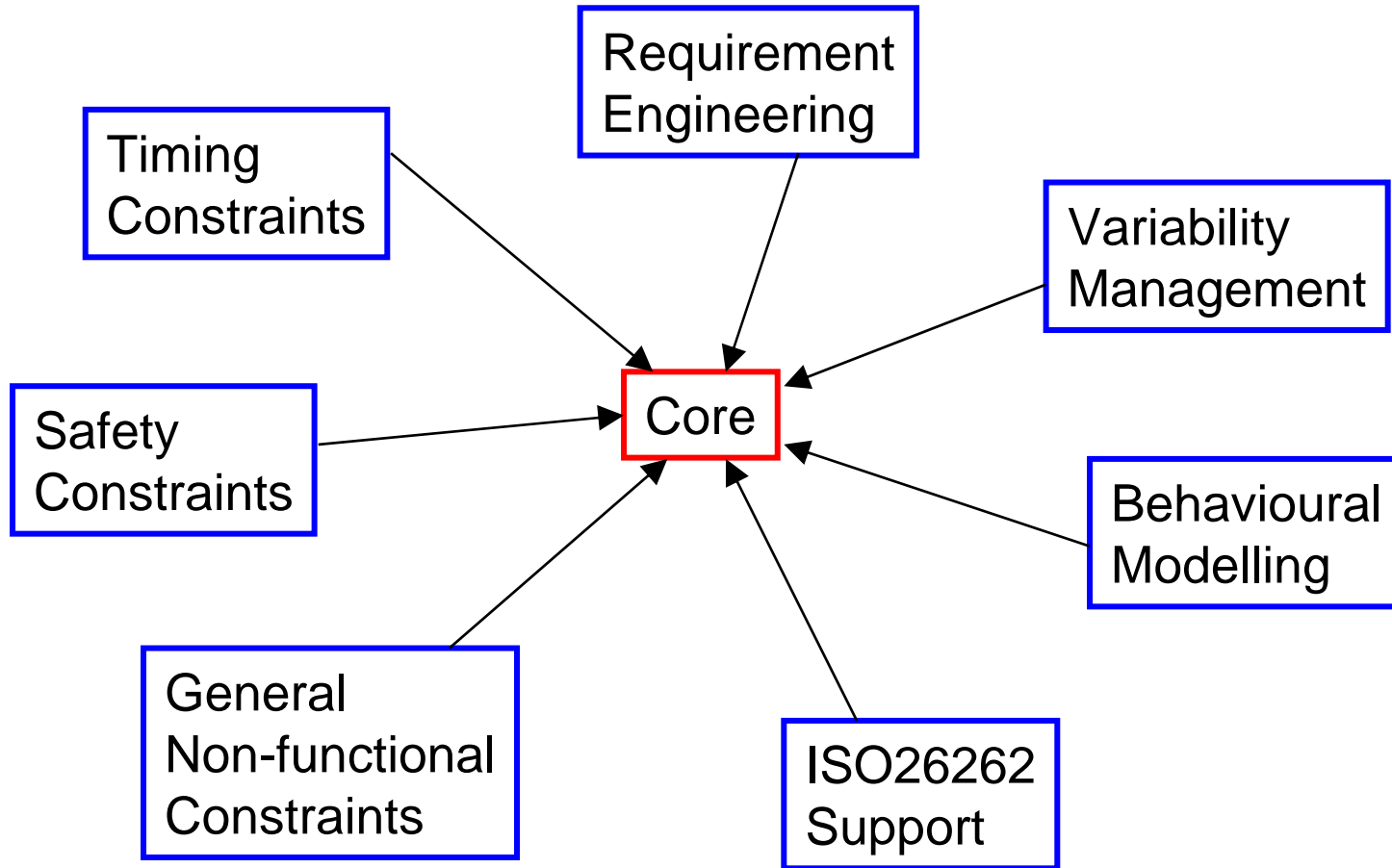
Examples of function-to-component Mappings



Function to runnable

The Information Scope Difference **ATESST**

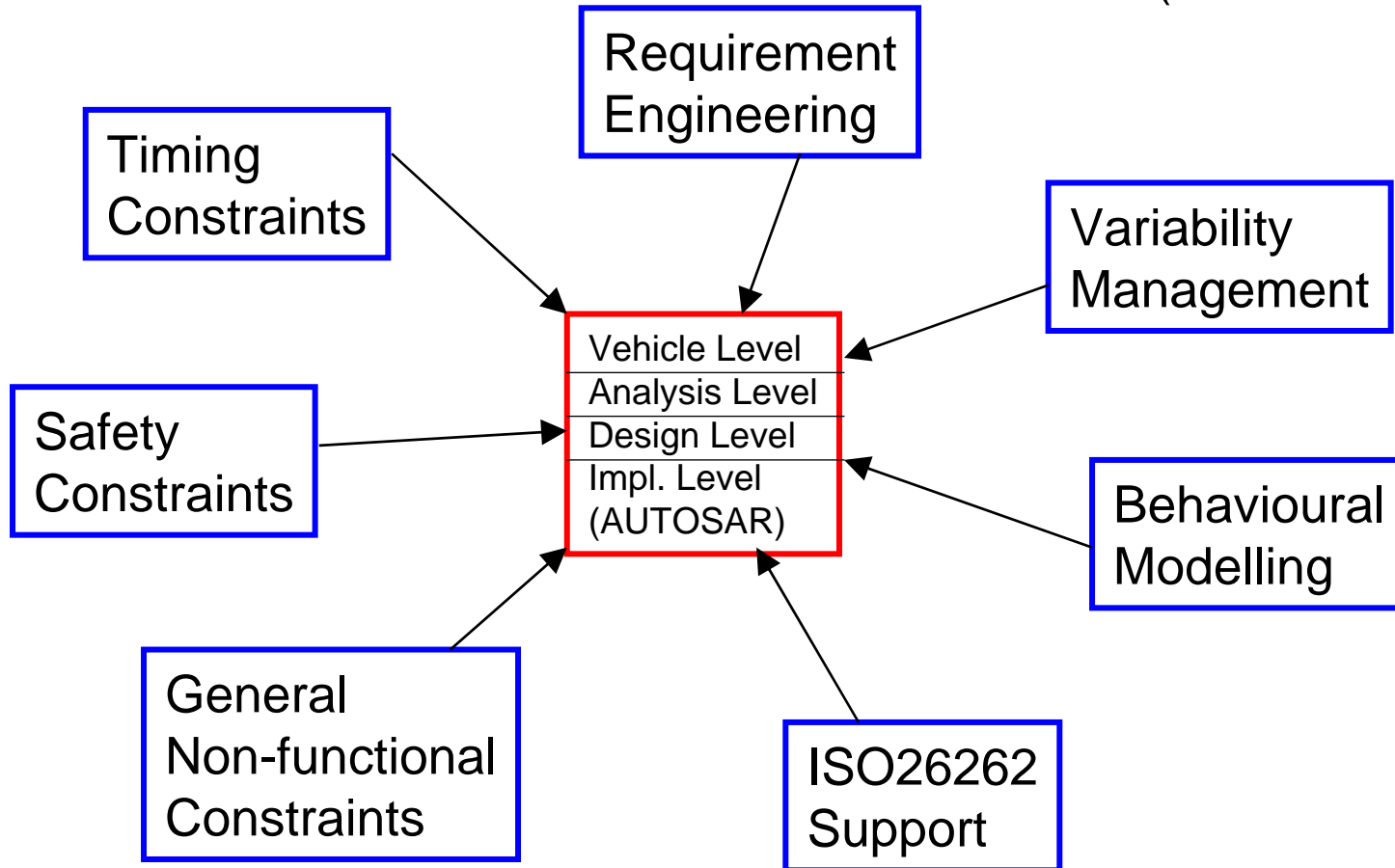
A Modular View of EAST-ADL



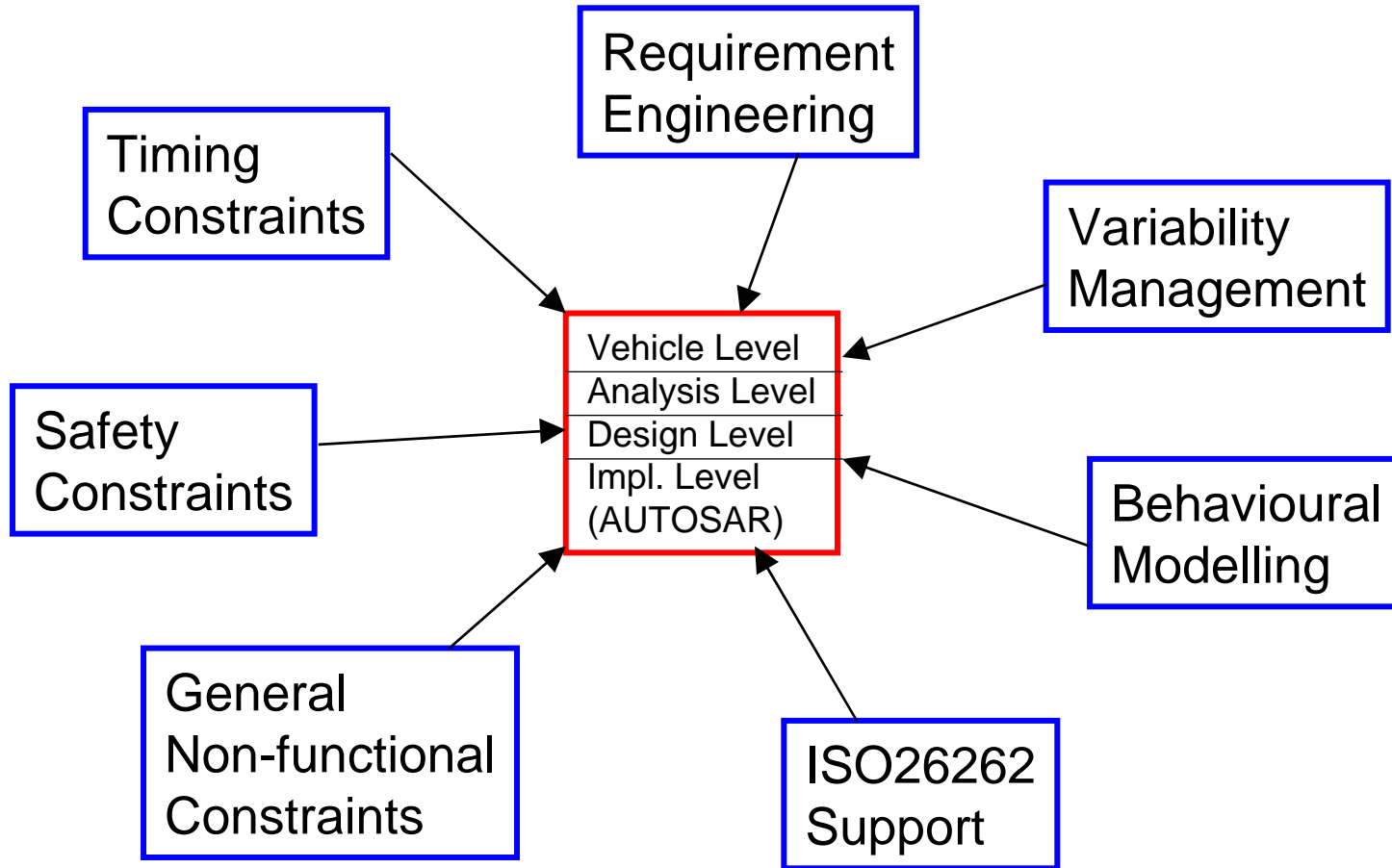
Adding Capability to EAST-ADL

⇒ Adding also to AUTOSAR

(to a certain extent)



Today: EAST-ADL fully AUTOSAR 3.x compatible enabling much of extensions to AUTOSAR 3.x templates



Conclusions

Abstraction Levels /Separation of Concerns

- AUTOSAR
 - **Defines the software architecture**
 - **Implementation details**
- EAST-ADL
 - **Design Level defines the functional architecture**
 - **Logic/Functional aspects**

Concepts / Information Scope

- AUTOSAR
 - **Limited scope**
 - **Higher in later versions**
- EAST-ADL
 - **Broader scope**
 - **Capable to Enrich AUTOSAR on Implementation Level**