

Software engineer defines SW Architecture

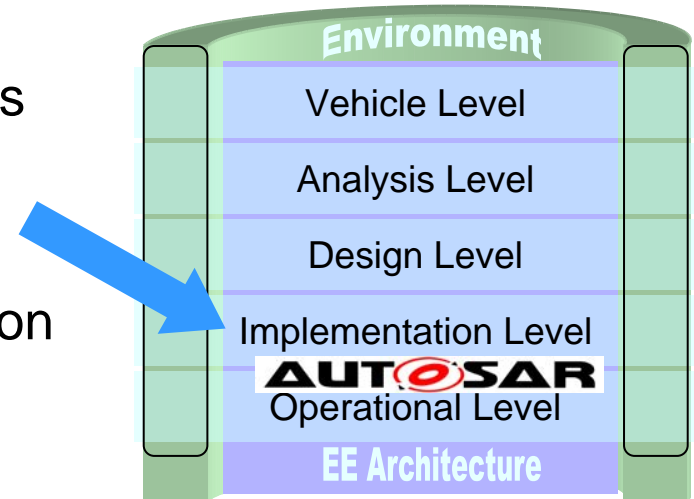


AUTOSAR Application SW Components are defined based on Design Level contents

The set of SW components together realizes the Functional Architecture

Software organization and functional organization is decoupled and optimization of the SW architecture is possible.

Legacy, sourcing, allocation, performance, verification, responsibility, re-use, etc. influence which functions are realized by each SW component



General

EAST-ADL is designed to support the engineering effort for automotive embedded systems

AUTOSAR is used to capture the software architecture

- the static view of the exchange between SWC and integration of SW components.

EAST-ADL complements AUTOSAR

- Autosar corresponds to the implementation level of EAST ADL
- Design level and up captures the "contents" of the implementation
 - Functional definition
 - Features
 - Variability
 - Requirements
 - V&V aspects
 - Safety aspects

Software vs. Functional Architecture

Software and functional architecture are 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

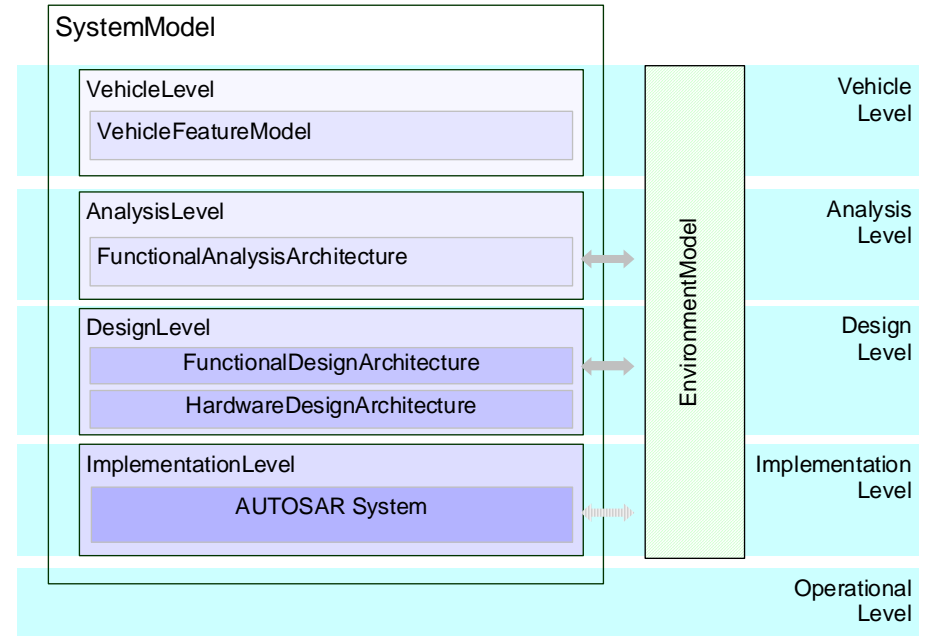
EAST-ADL concepts capture information that is the rationale for an implementation description using AUTOSAR concepts

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



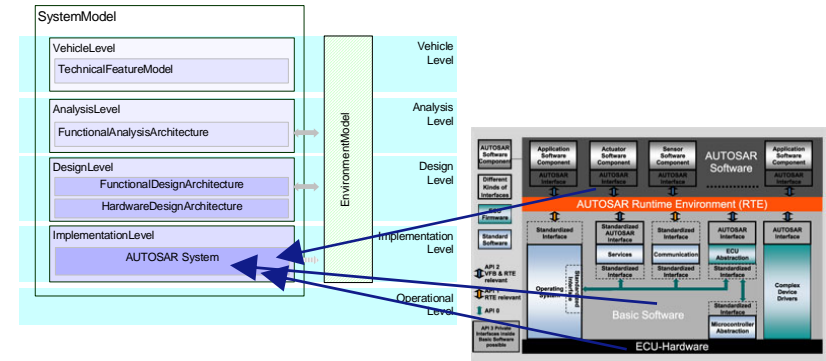
AUTOSAR defines

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



Mapping of EAST-ADL vs. AUTOSAR

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

DesignFunction represents structure and behavior in EAST-ADL

=> Fundamental mapping is 'DesignFunction – Runnable'

Different mappings are possible

- 'DesignFunction – AR Composition'
- 'DesignFunction – AR SW Component'

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

Function-to-Component mapping

Detailed mapping:

DesignFunction mapped to Runnable

- When a behaviorally and structurally precise relation is needed
- Rationale:
(elementary) DesignFunction is the non-concurrent “leaf” of EAST-ADL
Runnable is the non-concurrent entity in AUTOSAR
- N DesignFunctions to 1 Runnable

Black box mapping:

DesignFunction mapped to SW Component

- When Runnables are not exposed or detailed mapping is not needed
- N DesignFunctions to M SW Components
- DesignFunction and SW component may be composite or atomic

Port Mapping

Ports:

EAST-ADL FlowPorts mapped to AR Port-Interface-Dataelement

Exact mapping:

- 1 FlowPort to 1 DataElement

Free mapping

- N FlowPorts to N DataElement in M AR Ports

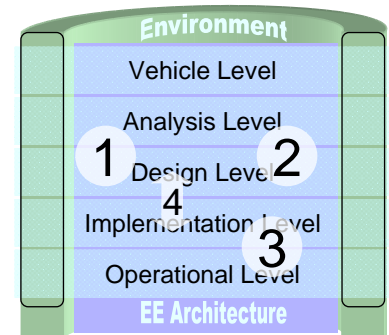
A possible methodology

Define functional structure in EAST-ADL

Optional: Identify blocks that should go together in a SW component and mark composites as "isComponent"

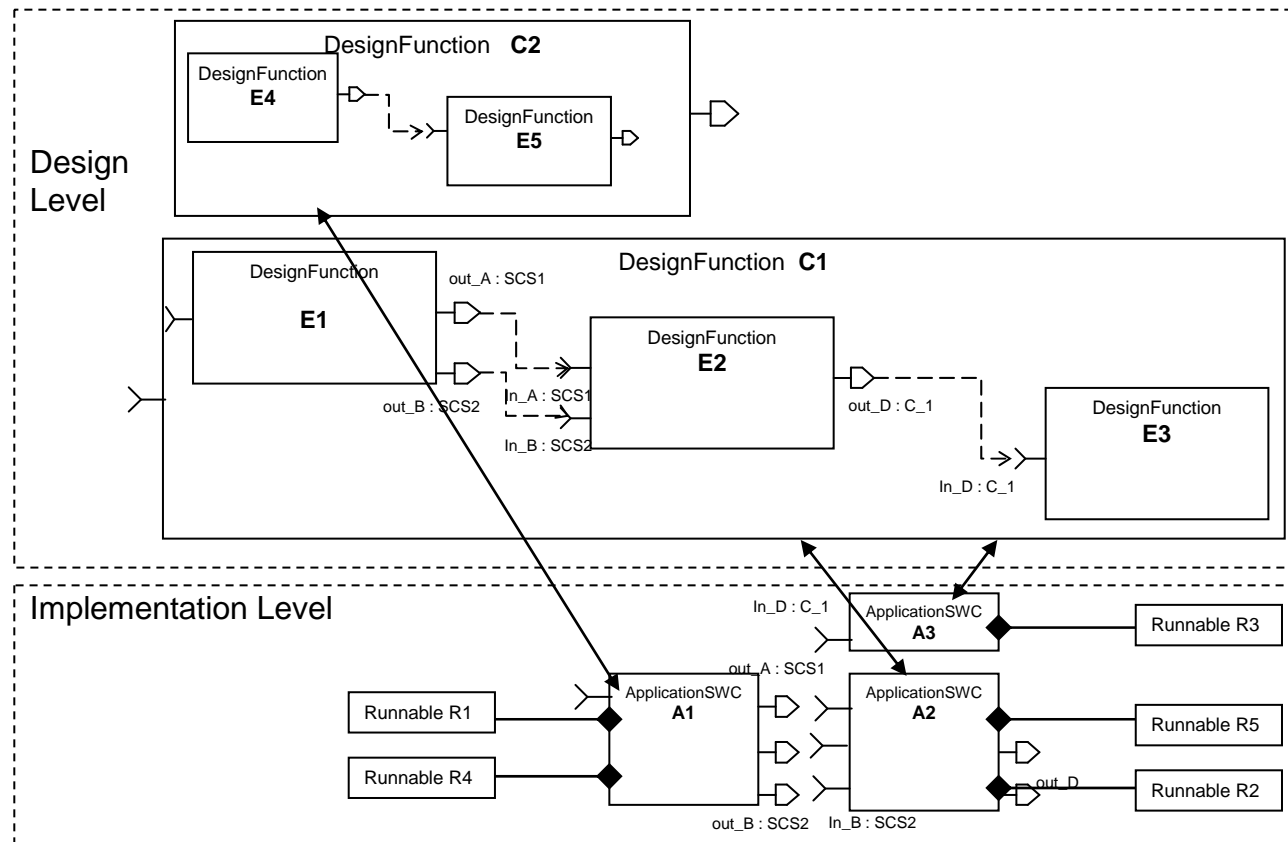
Define AR SWC and Runnables

Map elementary or composite DesignFunctions to appropriate AR SWC or Runnable (Realization relation)



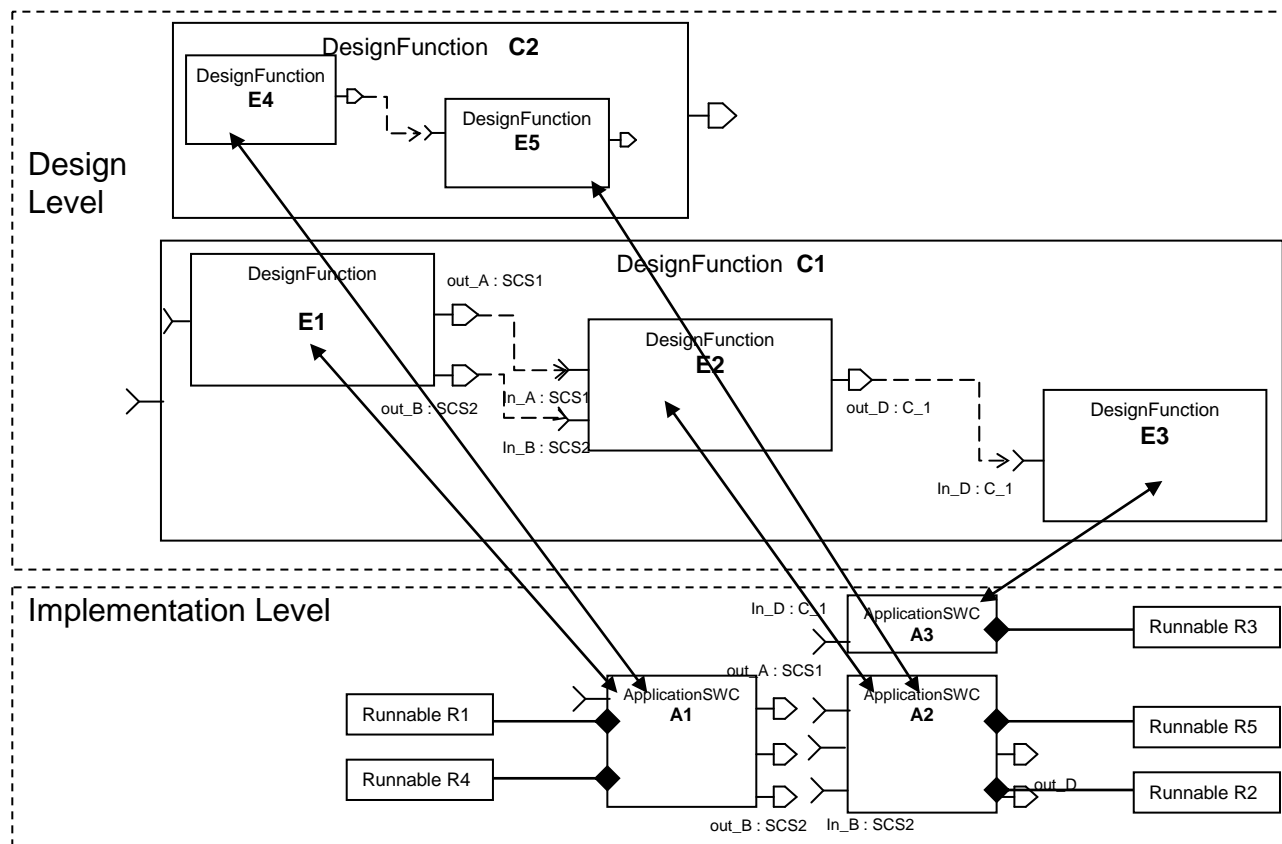
- the behavior of the runnable is defined in the corresponding elementary DesignFunctions
- the packaging into SWC/runnables is independent of functional structure
- SW architecture can be traced back to functions, features, requirements

Examples of Function-to-component Mappings



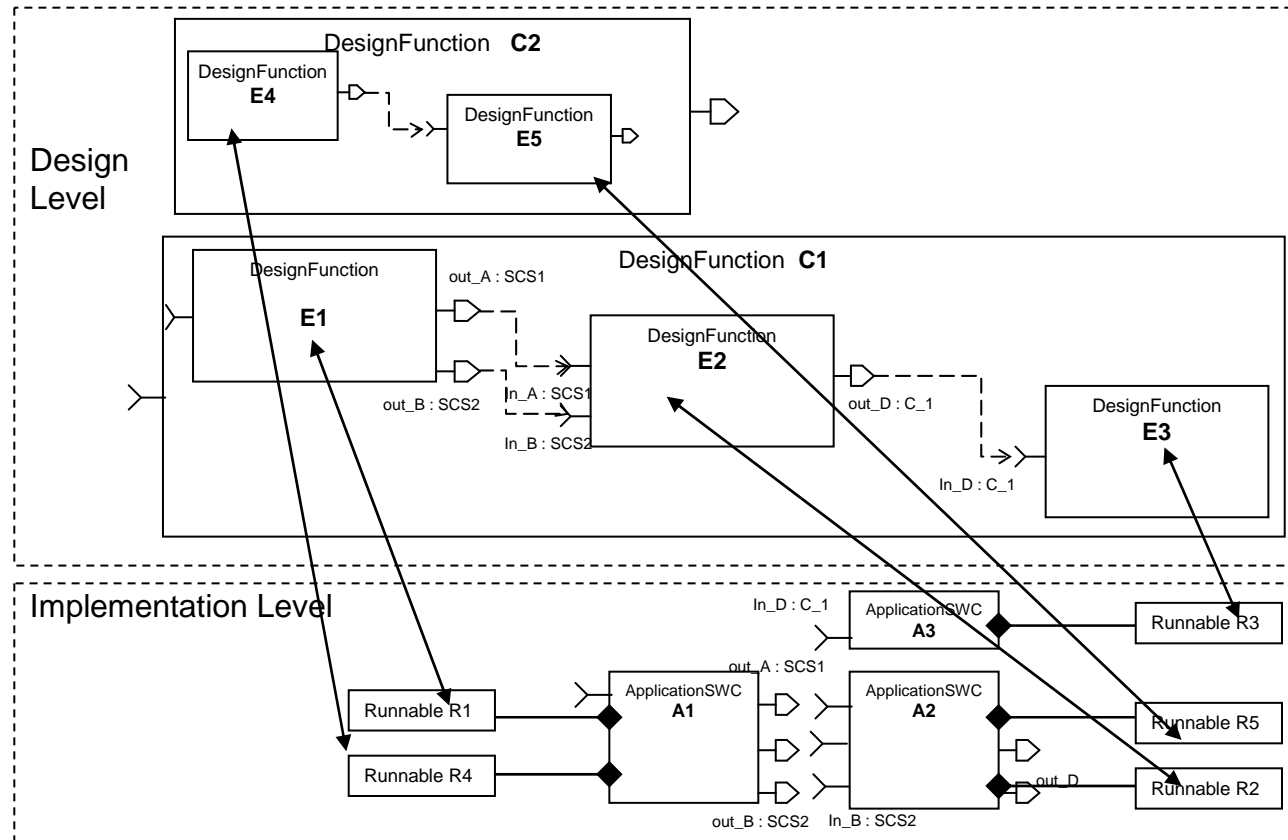
(non elementary) DesignFunction to SW Component

Examples of function-to-component Mappings



n (elementary) DesignFunction to 1 SW Component

Examples of function-to-component Mappings



(elementary) DesignFunction to runnable

Potential concerns

Why add Functional Design Architecture (FDA) as a separate model instead of adding functional definitions, timing and sequence, etc. to the AR atomic swc

- **Separation of concerns: software vs. function**
- **Provide a degree of freedom between the software architecture and the function architecture**
- **Timing, sequence, requirements, etc. for software are specific for the implementation**

Why have Design Level when Analysis Level can be used for functional definition

- **Analysis level is an abstract description of algorithms. It does not impose detailed solutions/implementation constraints**
- **Co-simulation and detailed analysis of SW vs. function cannot be done**

Summary – Implementation Level

AUTOSAR defines the software architecture – the final implementation

EAST-ADL Design Level defines the functional architecture

2 representations support separation of concerns:

Implementation details can be addressed in the AUTOSAR templates

Logic/Functional aspects can be addressed in EAST-ADL constructs