BOEING

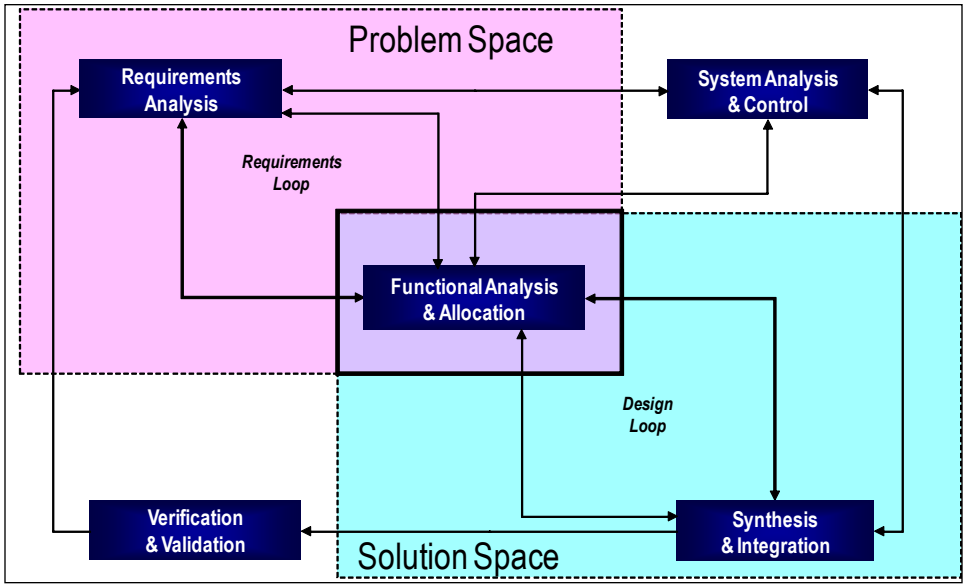
INCOSE International Symposium
Philadelphia, PA
June 24-27, 2013

Functional Architecture as the Core of Model-Based Systems Engineering

Dr. Ron Carson, INCOSE Fellow
Barbara J. Sheeley
The Boeing Company

Outline

- Introduction and motivation
- Role and benefits of functional architecture throughout the life cycle
  - From customer need statement to system architecture definition
  - Integration, Verification, Validation
  - Operations & Maintenance
- Summary



Functional Architecture is at the intersection of the Engineering problem and solution spaces

Introduction and Motivation

- SE is focused on system behavior (i.e., functionality)
- The requirements associated with the system mission are primarily associated functions and related performance.
- Derived system elements and interfaces must be shown to satisfy the requirements
- Integration is focused on the capability of the integrated system to perform system functions and mission threads
- Verification and Validation prove out “what it does”

Boeing is emphasizing *functionality* in *architecture*

What is a Functional Architecture?

- "An arrangement of functions and their subfunctions and interfaces (internal and external) that defines the execution sequencing, conditions for control or data flow, and the performance requirements to satisfy the requirements baseline." (IEEE 1220-2005, 3.1.15)
- Concepts
  - Functions decomposable into subfunctions
  - Interfaces
  - Control flows (sequences and decision logic)
  - Data flows (information exchanges)
  - Associated requirements
  - Satisfaction of requirements baseline

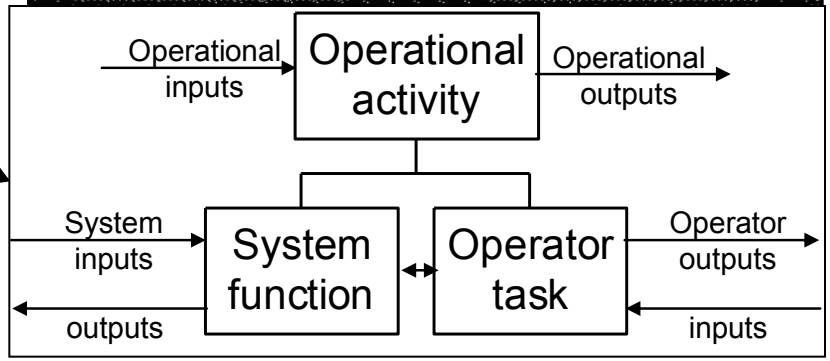
Elements of Systems Engineering Impacted by the Functional Architecture

- From Customer Need to System Architecture Definition
  - Requirements Definition
  - Requirements Analysis
  - Fault Analysis
  - Trade Studies
  - Interface Derivation, Analysis & Control
  - Architecture Allocations
- IV&V
  - Planning and Procedures
  - Risk, Issue, Opportunity Management
- Operations & Maintenance Development
  - Operator Procedure Allocations
  - O&M Issues & Opportunity Management

Functional Architecture impacts all elements of system definition

Stakeholder Requirements Definition

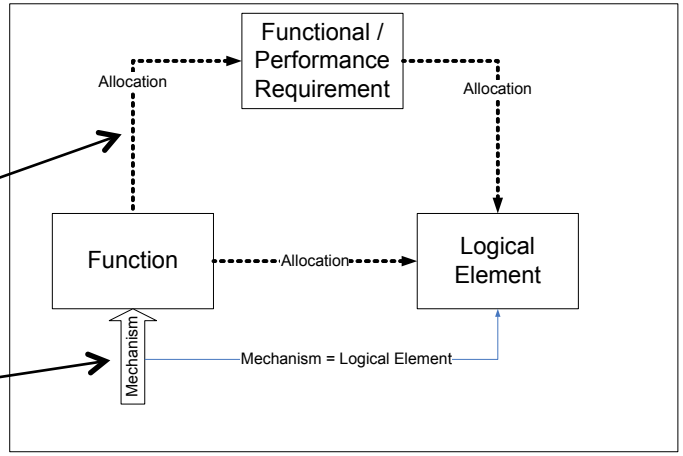
SE Process	Benefit from Availability of Functional Architecture	Consequences if Functional Architecture is Not Available
Requirements Definition	Derive requirements from functional architecture and MOEs – operational analysis	Requirements not fully justified; no connection to concept of operations
	Clearly delineate operator task from system functions	Unclear human-machine interfaces, functional interaction. Duplicate work during Training development



Decomposition of operational activity

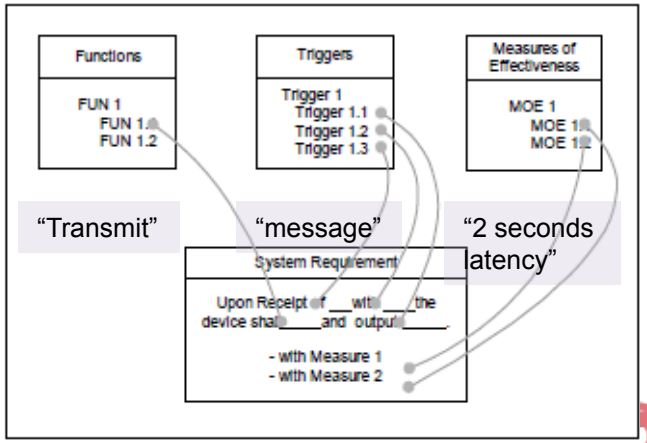
Requirements Analysis

SE Process	Benefit from Availability of Functional Architecture	Consequences if Functional Architecture is Not Available
Requirements Analysis	Identify all functional requirements (completeness)	Missing requirements
Requirements Analysis / Architecture	Find inconsistent or missing allocations of functions to mechanisms	Gaps and conflicts in functionality; rework.
Requirements Analysis	Identify verbs in functional requirements	Functional requirements not justified
	Identify states and modes as conditions on requirements	Required or Prohibited functions not identified



Finding missing functions, and allocations of functions to mechanisms

Mode → / Function ↓	Climb	Descend	Cruise	Survey
Lift aircraft	Required	Required	Required	Allowed
Propel aircraft	Required	Required	Required	Allowed
Display objects	Allowed	Allowed	Allowed	Required
Extend landing gear	Prohibited	Required	Prohibited	Allowed
Retract landing gear	Required	Prohibited	Required	Allowed

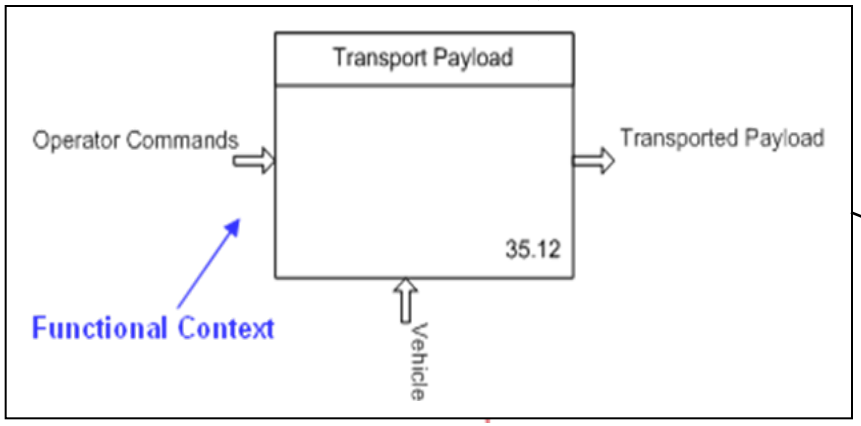


Defining verbs for functional requirements (after Piraino et al., 2001)

Requirements Analysis – Boundaries and Context

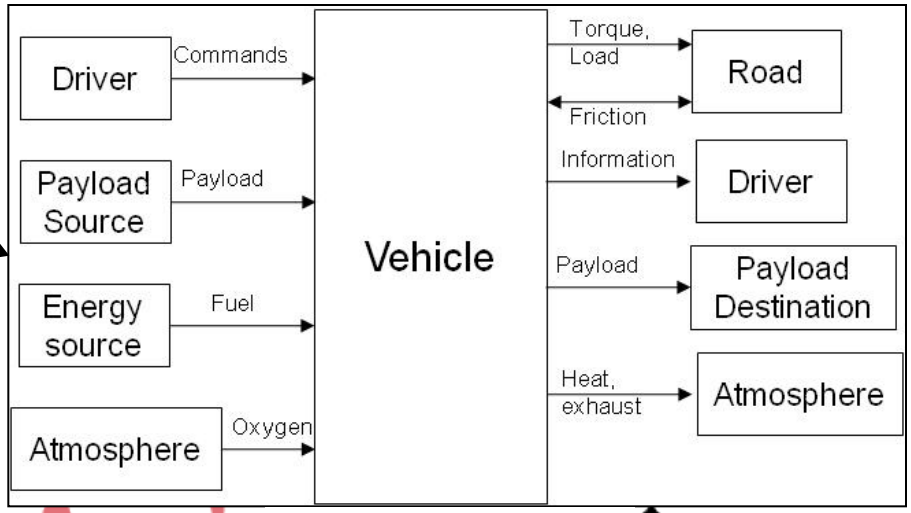
SE Process	Benefit from Availability of Functional Architecture	Consequences if Functional Architecture is Not Available
Requirements Analysis	Identify system boundary and external interfaces	Unclear or disputed boundary, context, and interfaces.

Identifying mission functions



IDEF0 A-0 functional context

Defining external interfaces from context



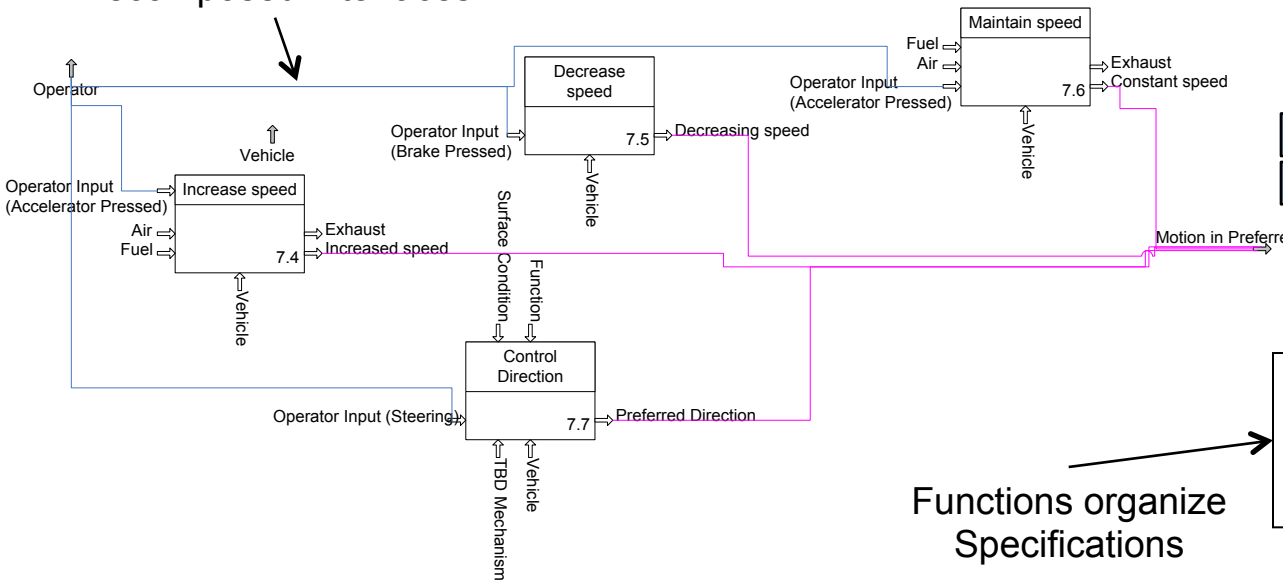
System context

Requirements Analysis – Interfaces and Specifications

SE Process	Benefit from Availability of Functional Architecture	Consequences if Functional Architecture is Not Available
Requirements Analysis	Decompose interfaces earlier for individual requirements	Late discovery of interface details
	Organize specifications by operations from FA	Requirements not clearly connected to operations

3.1.1.1	Operator Command - Accelerator
3.1.1.2	Operator Command - Brake
3.1.1.3	Operator Command - Steer

Decomposed Interfaces



Interfaces

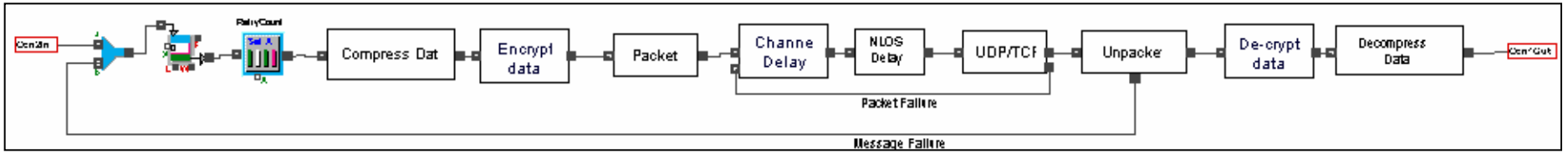
Requirements

Functions organize Specifications

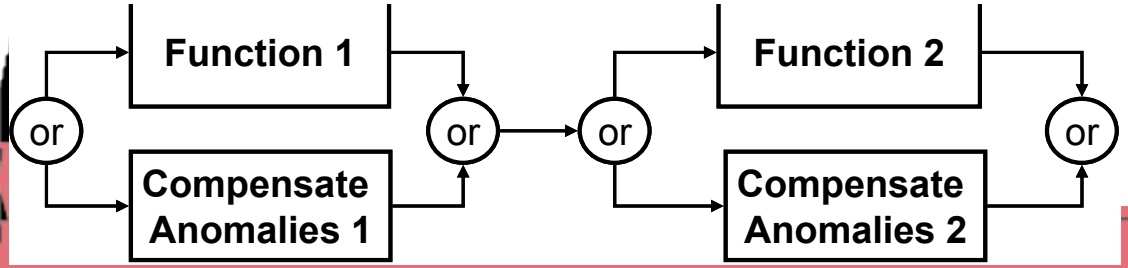
3.2.1.1	Increase speed
3.2.1.2	Maintain speed
3.2.1.3	Decrease speed
3.2.1.4	Control direction

Fault Analysis

SE Process	Benefit from Availability of Functional Architecture	Consequences if Functional Architecture is Not Available
Requirements Analysis	Enable analysis of nominal and off-nominal behavior	Failure to identify off-nominal conditions until integration or later → rework
Requirements Analysis, Architecture	Early "what if" analysis; model-based fault injection	Failure to identify incomplete or inconsistent requirements or architecture until integration or later → rework
Architecture	Identify functionality hazards (unintended behavior) and response to failure (fault detection, isolation & recovery)	Delayed discovery until integration → rework
	Executable – examine combinations of failure conditions	FMEA (single failure) or Fault Tree (top down based on end-effects and assumed contributions)

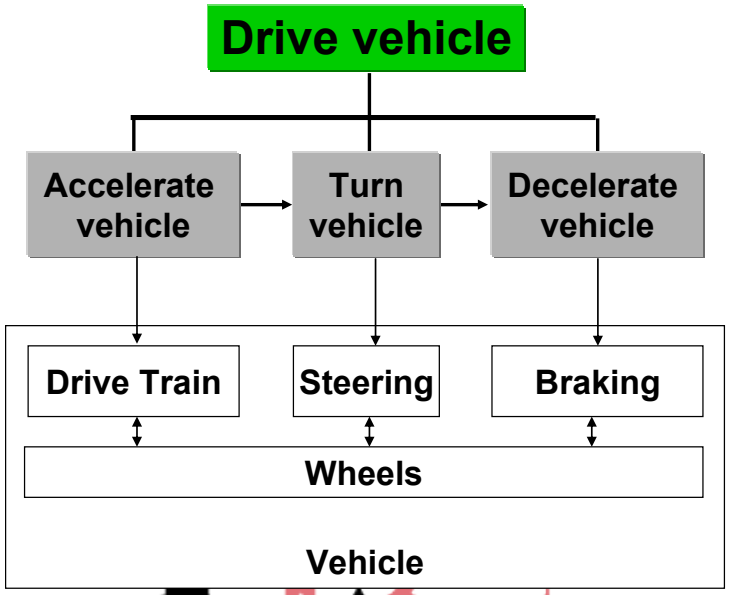


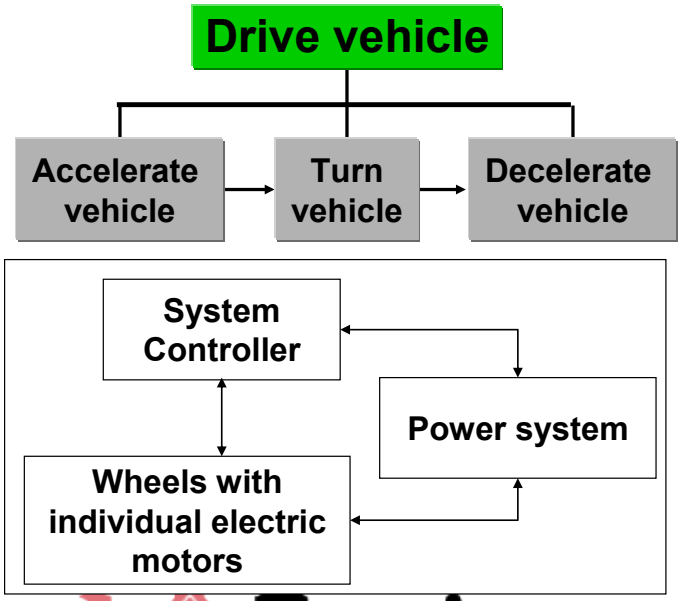
Executable model of communication channel for hazard and failure analyses



Trade Studies

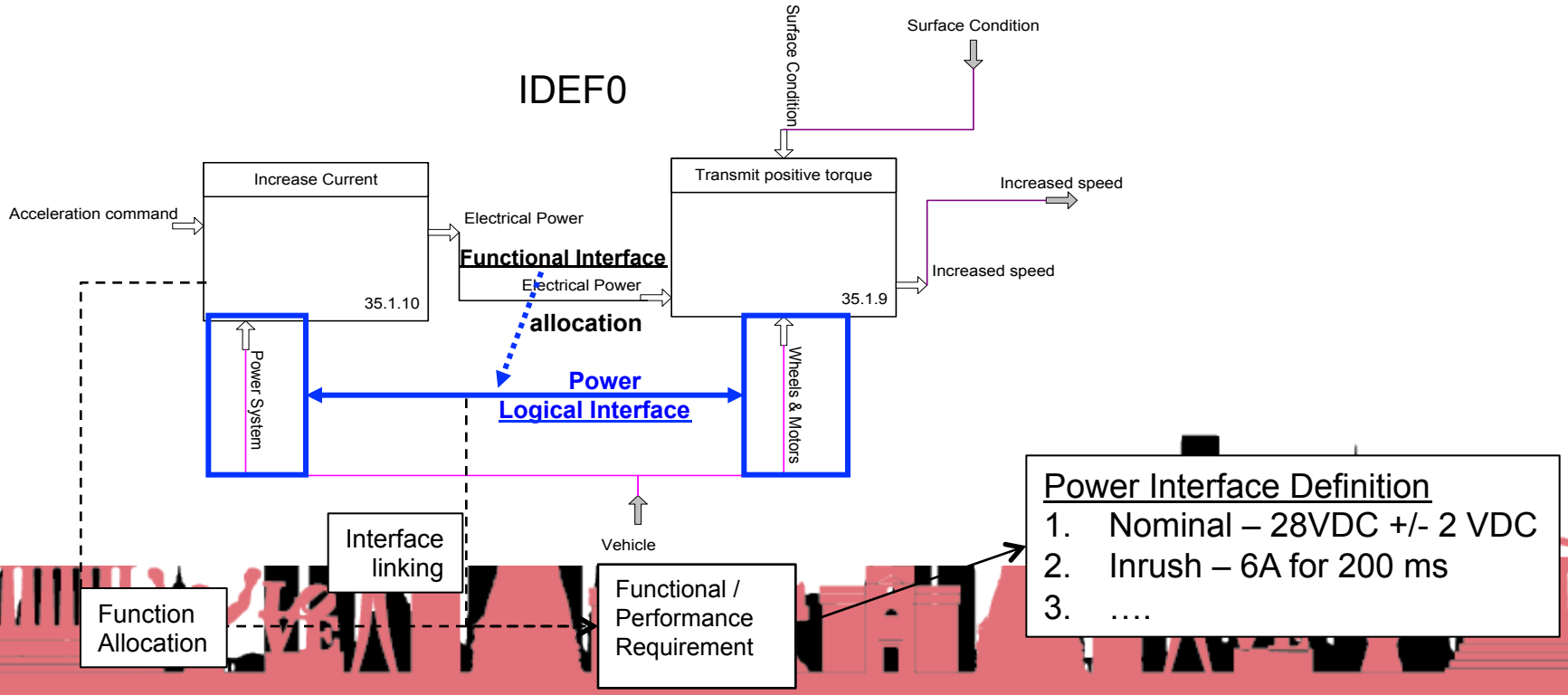
SE Process	Benefit from Availability of Functional Architecture	Consequences if Functional Architecture is Not Available
Trade studies (alternate architectures)	Optimize architecture vs. measures of effectiveness and suitability (cost effectiveness, design for value)	Unoptimized architectures – less value





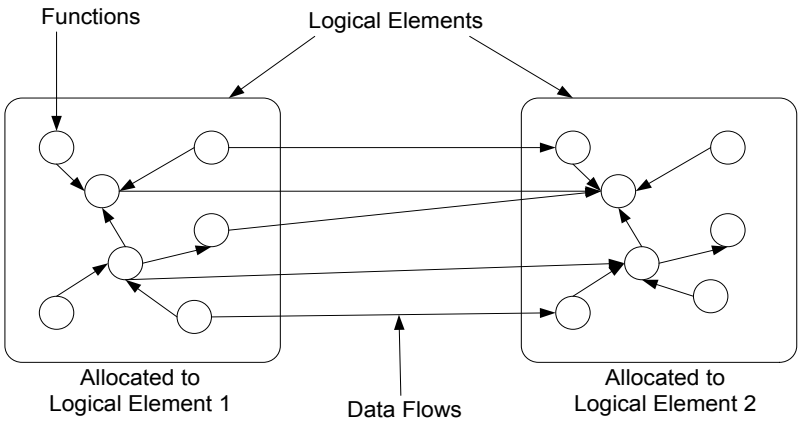
Interface Derivation

SE Process	Benefit from Availability of Functional Architecture	Consequences if Functional Architecture is Not Available
Architecture / interfaces	Derive logical interface from functional interfaces for different Mechanisms	Gaps and conflicts in interfaces; rework during IV&V
	Link interface definition to functional requirement; completeness of requirement	ICDs disconnected from Requirements (no link); duplicate work in IV&V (multiple tests of interfaces).

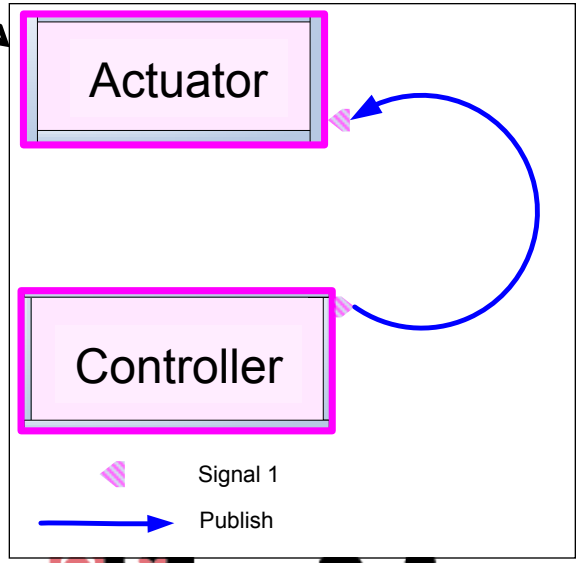


Interface Analysis

SE Process	Benefit from Availability of Functional Architecture	Consequences if Functional Architecture is Not Available
Architecture / interfaces	Early validation of interfaces and networks; find gaps, conflicts (publication/ subscription)	Delayed discovery during Integration & Verification → rework
	Understand / optimize interface coupling	Unoptimized interfaces; complexity, integration risk

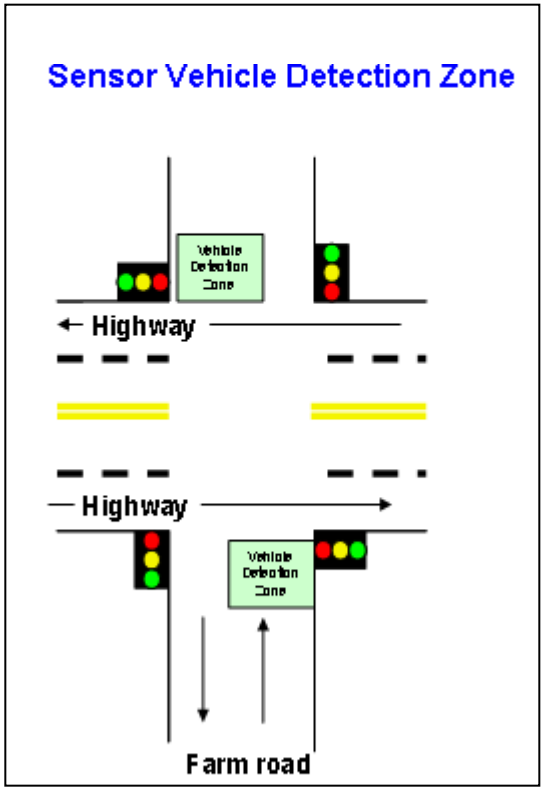


High interface coupling



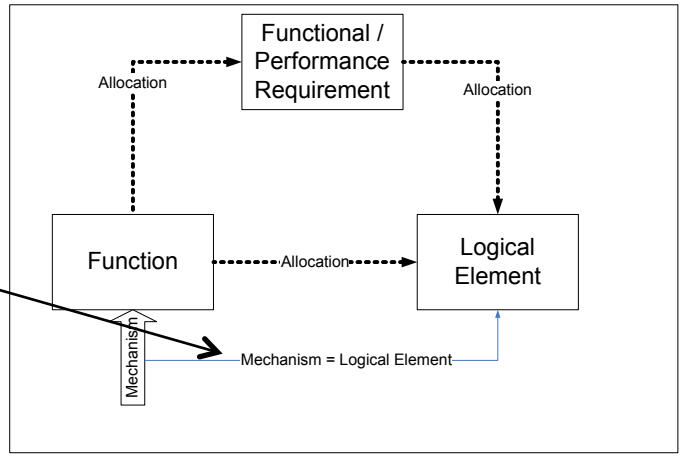
Interface Control

SE Process	Benefit from Availability of Functional Architecture	Consequences if Functional Architecture is Not Available
Architecture / interfaces	Publish ICDs directly from architecture	Manual generation or transcription; gaps and conflicts
	Manage changes within model (CM) for all affected ICDs	Version inconsistency, publication lag.

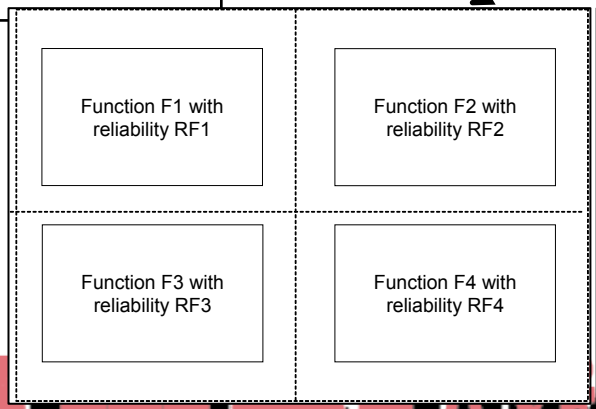


Allocations

SE Process	Benefit from Availability of Functional Architecture	Consequences if Functional Architecture is Not Available
Architecture	Clear association of behavior to logical elements	Unclear association of behavior to logical elements
	Validate reliability allocations top-down; drive change before detailed design	Bottom up; must live with result or rearchitect/ redesign with high program impact.



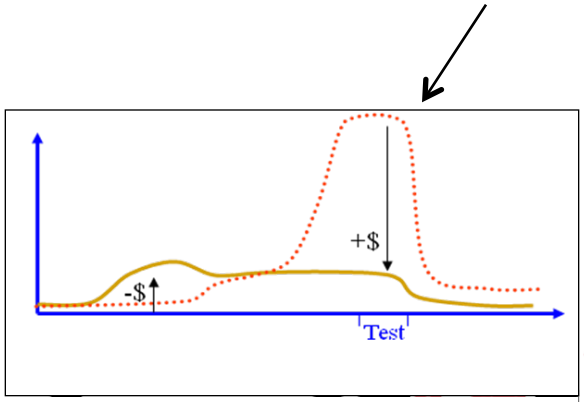
Allocate functions to logical elements



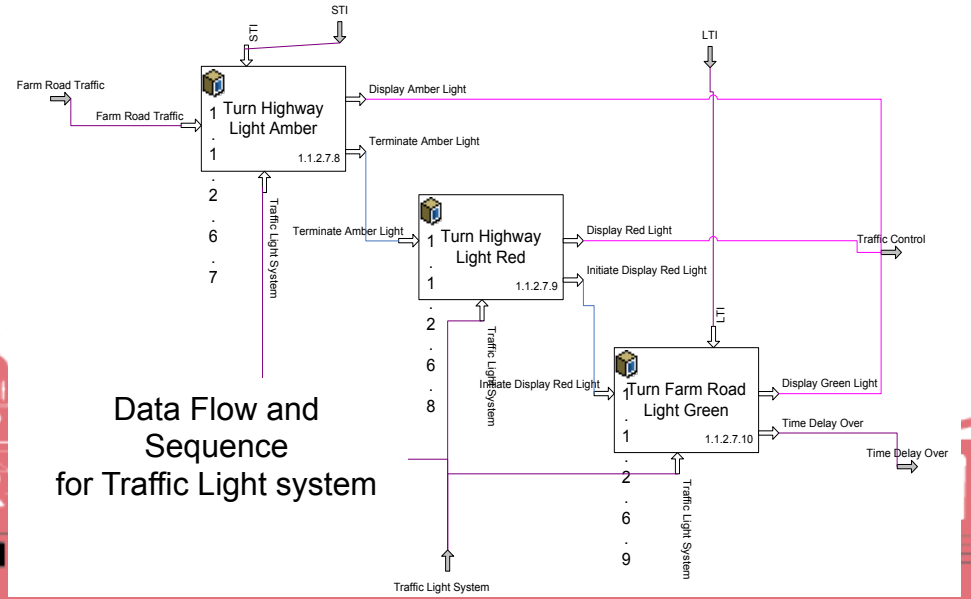
Function Reliability	
Function(n)	R(Fn)
F1	0.99900
F2	0.99800
F3	0.99990
F4	0.99000
Minimum R	0.986933
=Product(R(Fn))	

IV&V Planning and Procedures

SE Process	Benefit from Availability of Functional Architecture	Consequences if Functional Architecture is Not Available
Implementation	Autogenerate solution from functional model	Build HW and SW from non-model artifacts, with possible gaps and conflicts.
Transition, Integration & Verification	Generate use cases for planning the incremental build-up and deployment of logical elements and associated infrastructure.	Non-optimized integration build-up plan. Ad hoc planning from rediscovered functional threads.
Transition, Integration & Verification	Generate use cases for procedures	Gaps in integration checks; ad hoc sequencing from requirements.
Verification	Analytical verification using model	Analytical verification from “scratch” or testing, with higher costs



Relative program costs through Verification with (solid) and without (dotted) functional integration.

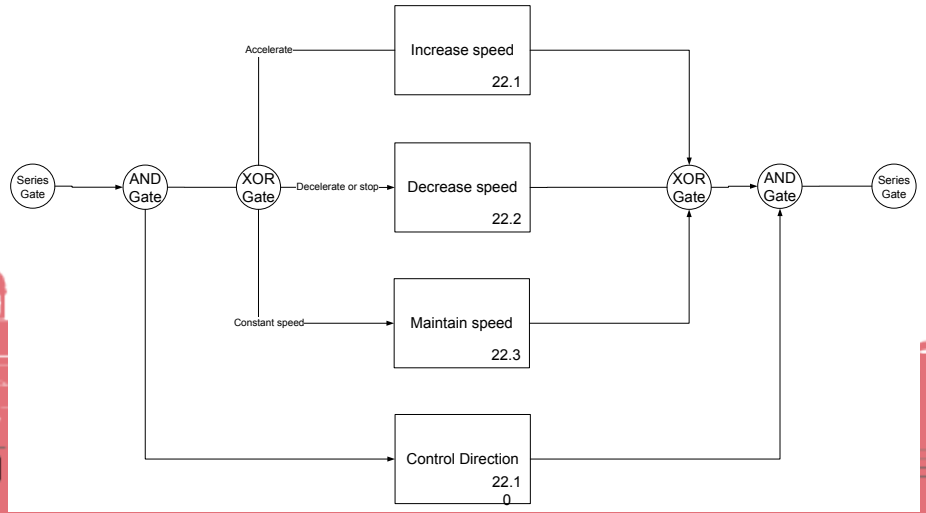


Data Flow and Sequence for Traffic Light system

IV&V Risk Management

SE Process	Benefit from Availability of Functional Architecture	Consequences if Functional Architecture is Not Available
Integration and Verification	Confirming expected behavior.	Discovering actual behavior; rework for the unexpected.
	Clear association of interfaces to functionality; effects of interface discrepancies can be quickly assessed for functional impact.	Cannot easily determine functional impact of interface discrepancies.
	Identify areas of integration risk due to concurrent functionality, critical timing relationships, and required ("must do") or prohibited ("must not do") state or mode dependencies	Delayed discovery until integration with rework, cost, schedule increase

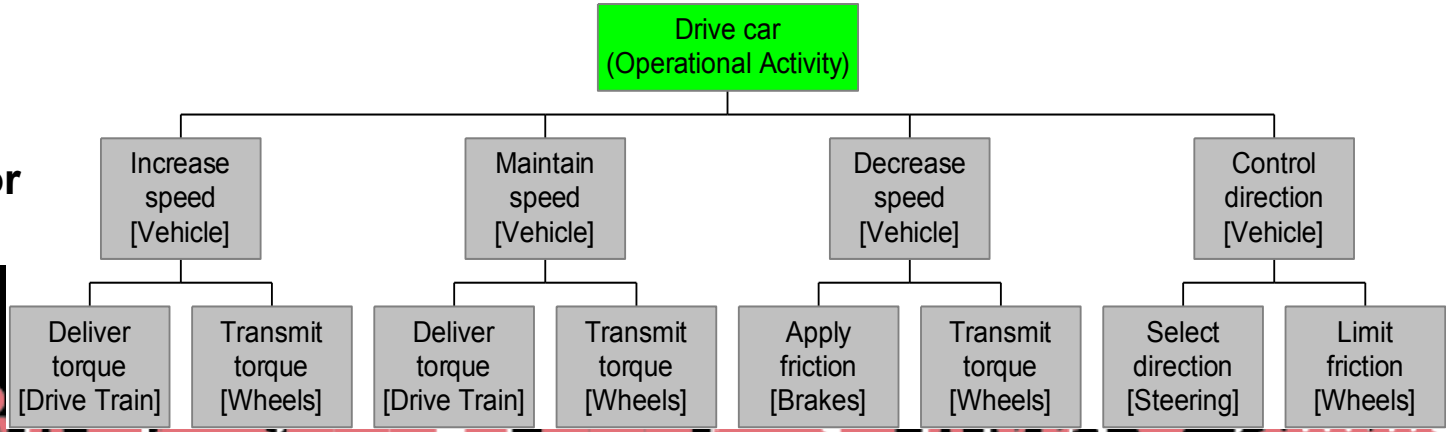
Functional concurrency for Road Vehicle operations



IV&V Issue Management

SE Process	Benefit from Availability of Functional Architecture	Consequences if Functional Architecture is Not Available
Integration and Verification	Aid replanning when logical elements not available	"Down time" waiting for equipment, or uncertain use of incomplete systems (unclear whether functions will work).
Verification	Provide information for functional/ performance reallocation for any deficiencies during Verification	Limited choices (deviation or fix design); more ad hoc brainstorming for reallocation.
	Anomaly resolution root cause substantiated by model; enable accepting "as is" as low-probability event	Fix anomaly based on symptoms.
Validation	Validate capability to satisfy user needs ahead of operational test & evaluation (OT&E)	Discover deficiencies during OT&E

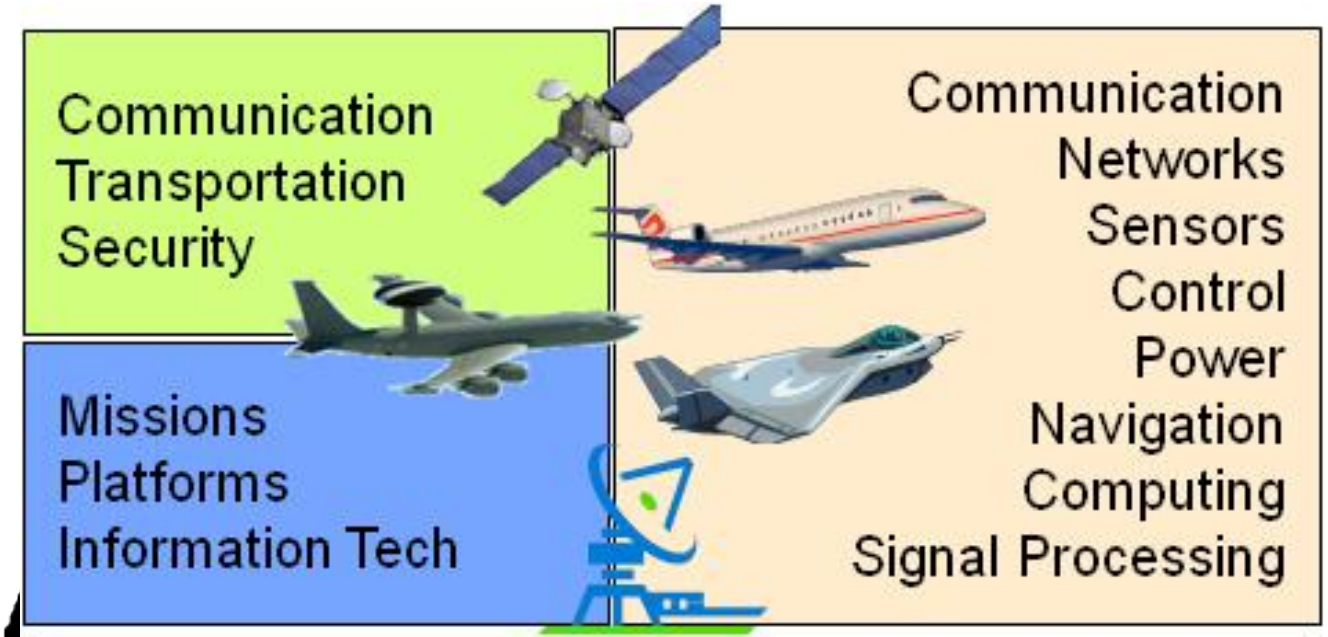
Functional decomposition for Road Vehicle



IV&V Opportunity Management

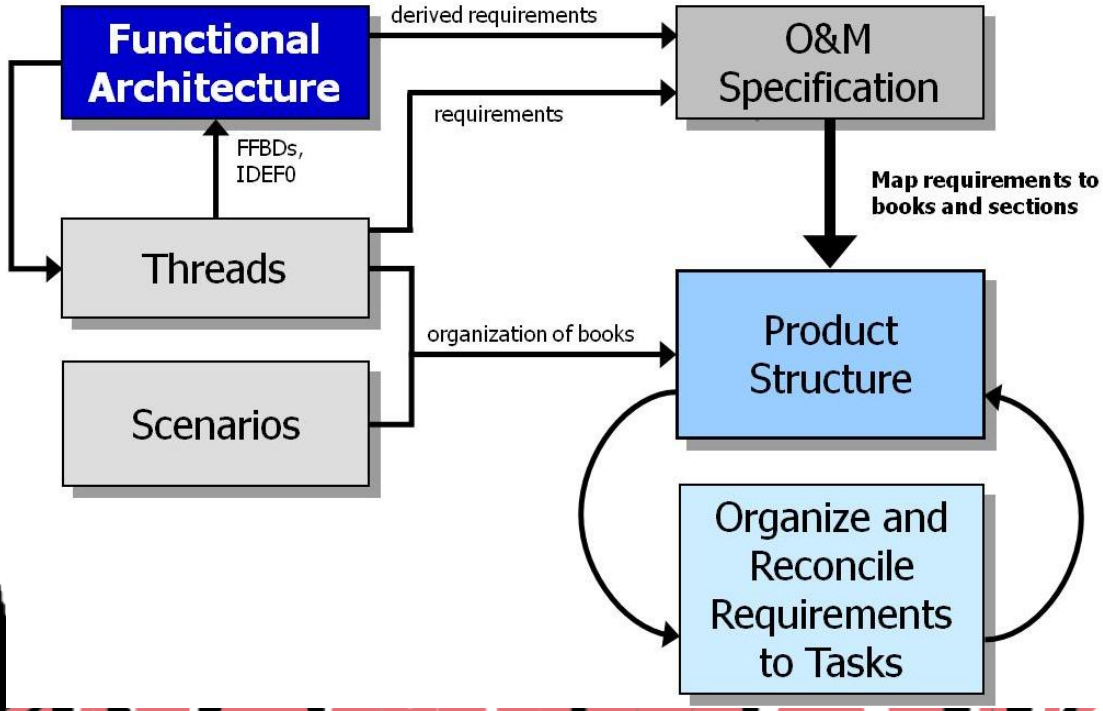
SE Process	Benefit from Availability of Functional Architecture	Consequences if Functional Architecture is Not Available
Validation	Ability to predict additional capabilities by analysis, ahead of operational evaluation	Require testing to discover additional capabilities.

System and SoS capabilities evaluated analytically – Virtual Experimentation



Operations & Maintenance Allocations

SE Process	Benefit from Availability of Functional Architecture	Consequences if Functional Architecture is Not Available
Operations & Maintenance	Define operator procedures from architecture (functional sequences of items allocated to "operator" or "maintainer")	Manually generate procedures based on system requirements, description and interfaces; resolve gaps and conflicts with designers



Operations & Maintenance Issues & Opportunities

SE Process	Benefit from Availability of Functional Architecture	Consequences if Functional Architecture is Not Available
Operations & Maintenance	Analyze operational problems (simulation) using functional model to identify root cause and validate solutions.	Use system hardware and software to determine root cause and validate solutions.
	Identify revised or new functionality allocations for (diminished manufacturing sources) DMS or technology insertion. Includes COTS refresh.	After-the-fact identification of gaps and conflicts in functions, interfaces, performance for DMS changes, technology insertion, and COTS refresh.
	Enable product line management with clear identification of gaps in new capabilities	Manage related systems in isolation with extra cost and configuration management challenges

Summary & Conclusions

SE Process	Benefit – Why do it?
Requirements Definition	Provides clear relationship of requirements to needs and operations
Requirements Analysis	Enables complete and consistent requirements, boundaries, interfaces, specifications, allocations
Trade Studies	Exposes hidden trade space; enables optimizing architectures for value
Fault management	Earlier analysis of functional failures and fault management
Architecture and Interfaces	Clear requirements allocations; consistent interface definition and management
IV&V Planning	Deployment and integration planning and procedures derived from functional allocations and use cases
IV&V Execution	Allows functional or interface impact of requirement non-compliance to be easily analyzed
Operations & Maintenance	Define operational procedures from architecture; analyze opportunities; resolve issues

Focusing on functional architecture yields benefits throughout the life cycle