Methodology-Driven MBSE:

Arcadia, Capella and Systems Modeling Workbench

Lisa Murphy INCOSE Huntsville, AL Chapter July 8, 2021

SIEMENS

Agenda

Methodology-Driven MBSE

Model functional-system architecture

Advanced systems management with product architecture

Re-use product architecture through multi-domain viewpoints

Close-the-loop with the product architecture

Methodology-Driven MBSE

Musing about Legacy

Trajectory of MBSE

…as SE-focused

… as Product Development Focused

… as Life cycle Focused

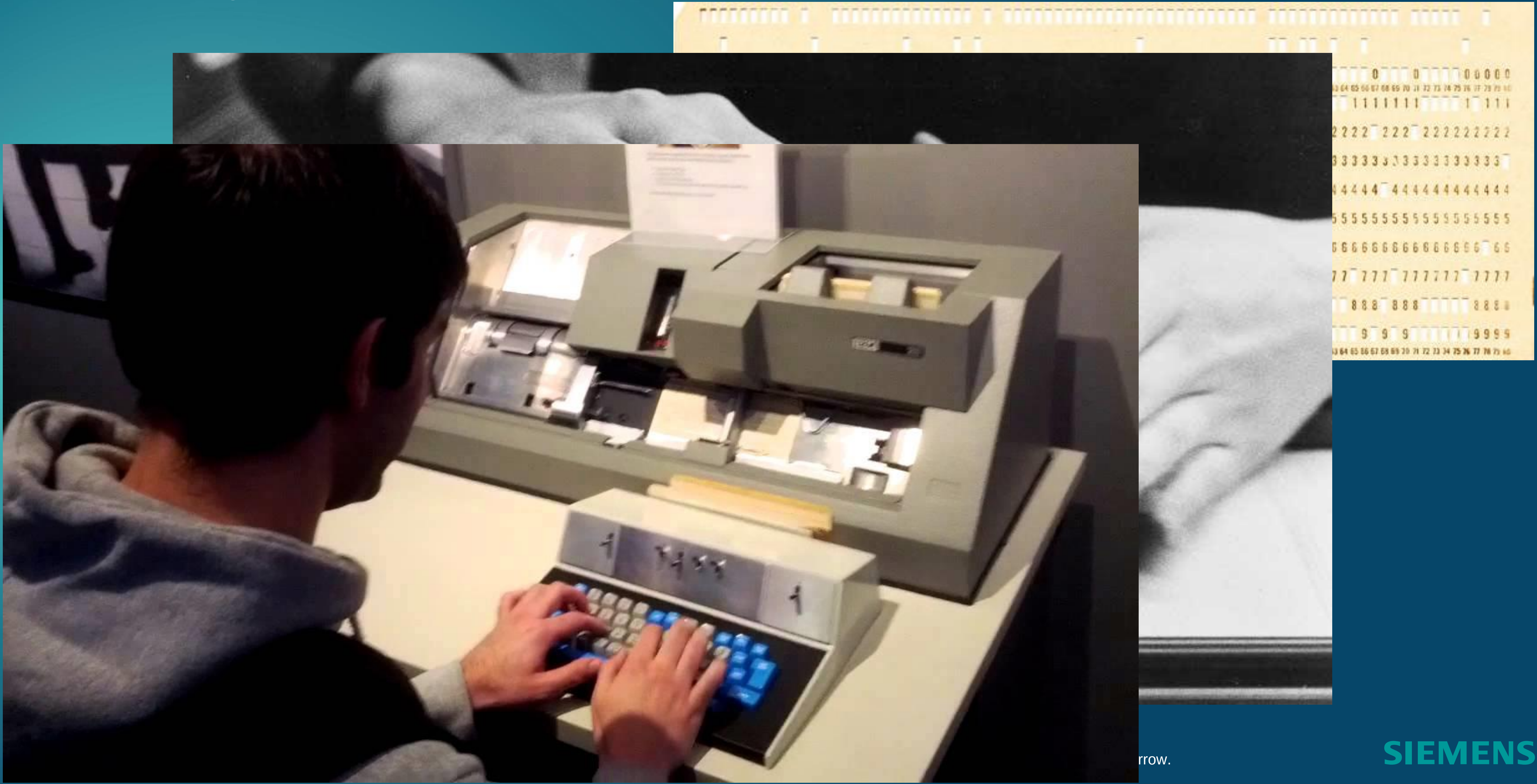
MBSE with method, tool, and clarity

ARCADIA

Capella

Systems Modeling Workbench

Lock-in, Legacy, or Backwards Compatibility?



rrow.

Who’s your customer and how do you know you are done?

Trajectory of MBSE

…as SE-focused

Systems Engineering

… as Product Development Focused

Developmental Engineering

… as Life cycle Focused

Everyone

While Seeking “Forward Compatibility”

MBSE with method, tool, and clarity

ARCADIA

Capella

Teamcenter Systems Modeling Workbench

The ARCADIA-Capella Journey

Thales (formerly Thompson)

- *French Commercial Aerospace & Defense Prime*
- *8th largest in world*

Wanted a Systems Engineering approach

Without needing modeling expert sitting with a domain SME

All stakeholders share same methodology, same information, same description of need and system in the form of a SHARED Model

Specialty needs met by viewpoints

Co-engineering between met by joint elaboration of models related in known ways across levels

Spent a few years working with SysML

They rejected much of it, kept some, and

developed an in-house method called ARCADIA

Learned and refined by using it in-house

Sought to share with partners, customers, suppliers

Developed in-house tool called Melody

With embedded modeling syntax and semantics

Released to the world as Capella via open source Eclipse Foundation project called PolarSys in 2014

Capella continues to advance core capabilities & number of adopters

Active Capella development community with SDK

Estimated worldwide user base of >400

Doubling in last few years; 10-15 new users/mo

Proven in real-time, real-scale, real-complexity projects

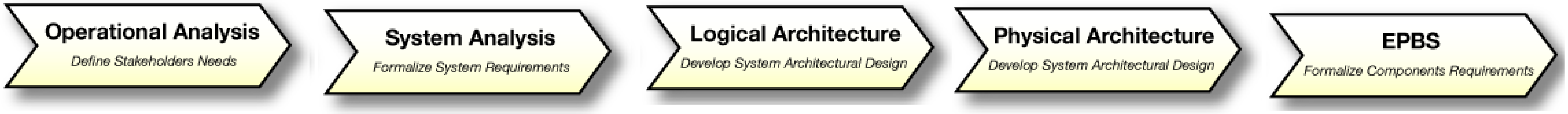
Thales alone has done several hundred non-trivial projects

They continue to be instrumental in open source

ARCADIA inspired new ISO 42030

The ARCADIA Methodology

ARCADIA (ARChitecture And Design Integrated Approach) is a system and software architecture engineering method, based on architecture-centric and model-driven engineering activities. A tool-supported approach to Define/ Analyze, Design and Validate System, Software, Hardware Architectures. It is adaptable to top-down, bottom-up and mixed applications via the Capella modeling tool.


Operational Analysis
Define Stakeholders Needs
System Analysis
Formalize System Requirements
Logical Architecture
Develop System Architectural Design
Physical Architecture
Develop System Architectural Design
EPBS
Formalize Components Requirements

What the (future) users of the system need to accomplish.
Clarify why capability is needed; system boundary explored but not set.

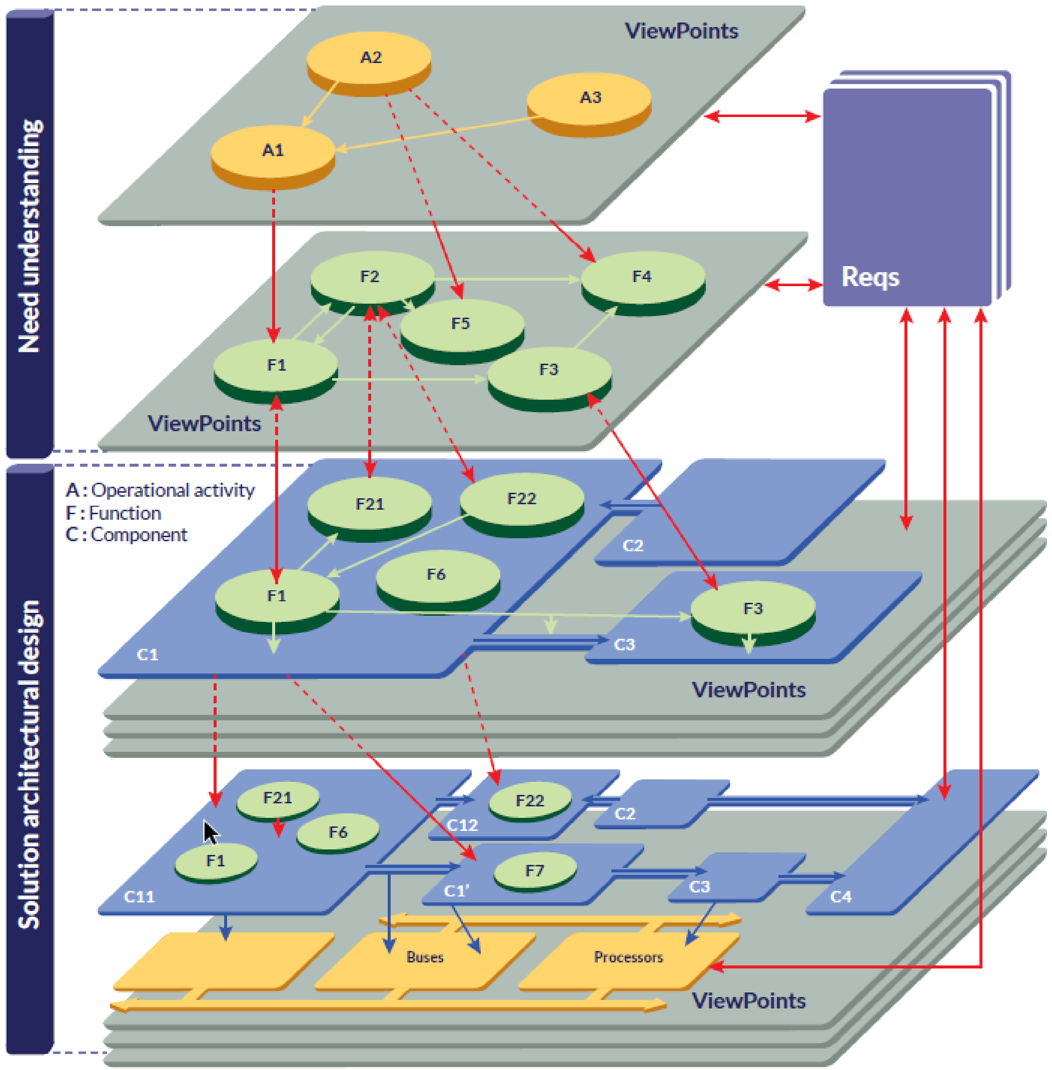
What the system needs to accomplish for the users.
System is “black box” here defining boundary and identifying functions needed.

How system will work to fulfill expectations, including non-functional constraints.
Decomposes System Functions into Logical ones and allocate them to Logical Components.

How system will be developed and built.
Identifies physical components to satisfy the conceptual design; decomposed logical functions are allocated to physical components. Adds functions required by implementation and technical choices.

What is expected from the provider of each component.
Captures make/buy and other build and implementation details [optional; least used].

Four Layers: Diagrams are representations of the same underlying model which knows the relationships



Operational Analysis
What the users of the system need to accomplish

Functional & Non Functional Need
What the system has to accomplish for the users

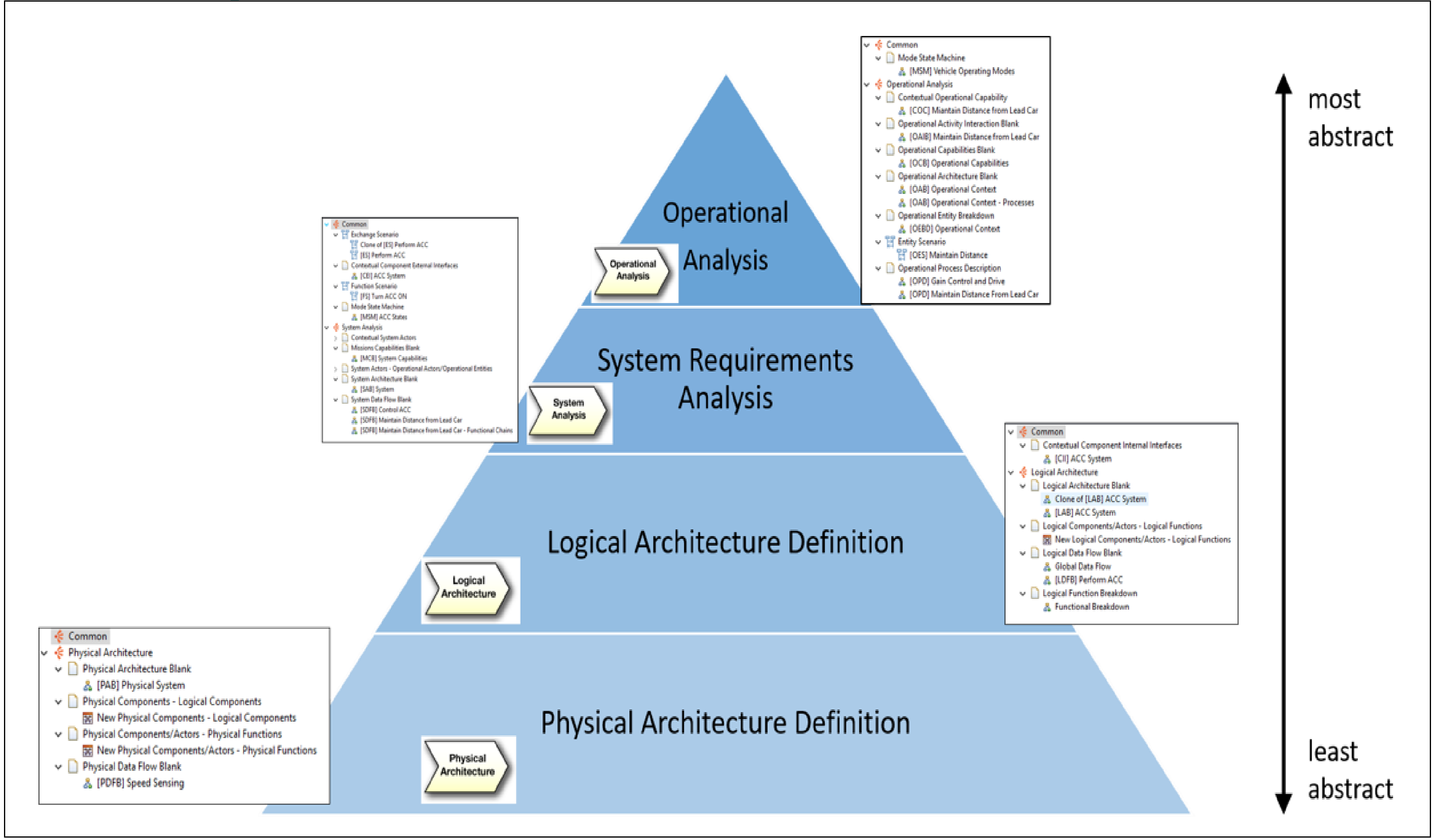
Logical Architecture
How the system will work to fulfill expectations

Physical Architecture
How the system will be developed and built

For clean-sheet development, start top-down, for extension, start bottom up with physical definition, reverse engineering logical architecture & functions

Red arrows show automatic traceability between similar objects as system definition evolves; other “modeling accelerators” reduce modeling effort.

Modeling Adaptive Cruise Control (ACC) with ARCADIA/Capella



- Common
  - Mode State Machine
    - [MSM] Vehicle Operating Modes
- Operational Analysis
  - Contextual Operational Capability
    - [COC] Miantain Distance from Lead Car
  - Operational Activity Interaction Blank
    - [OAIB] Maintain Distance from Lead Car
  - Operational Capabilities Blank
    - [OCB] Operational Capabilities
  - Operational Architecture Blank
    - [OAB] Operational Context
    - [OAB] Operational Context - Processes
  - Operational Entity Breakdown
    - [OEBD] Operational Context
  - Entity Scenario
    - [OES] Maintain Distance
  - Operational Process Description
    - [OPD] Gain Control and Drive
    - [OPD] Maintain Distance From Lead Car

- Common
  - Exchange Scenario
    - Clone of [ES] Perform ACC
    - [ES] Perform ACC
  - Contextual Component External Interfaces
    - [CEI] ACC System
  - Function Scenario
    - [FS] Turn ACC ON
  - Mode State Machine
    - [MSM] ACC States
- System Analysis
  - Contextual System Actors
  - Missions Capabilities Blank
    - [MCB] System Capabilities
  - System Actors - Operational Actors/Operational Entities
  - System Architecture Blank
    - [SAB] System
  - System Data Flow Blank
    - [SDFB] Control ACC
    - [SDFB] Maintain Distance from Lead Car
    - [SDFB] Maintain Distance from Lead Car - Functional Chains

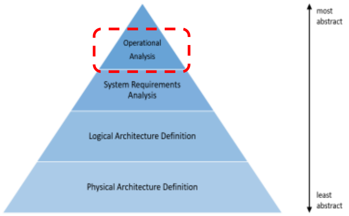
- Common
  - Contextual Component Internal Interfaces
    - [CII] ACC System
- Logical Architecture
  - Logical Architecture Blank
    - Clone of [LAB] ACC System
    - [LAB] ACC System
  - Logical Components/Actors - Logical Functions
    - New Logical Components/Actors - Logical Functions
  - Logical Data Flow Blank
    - Global Data Flow
    - [LDFB] Perform ACC
  - Logical Function Breakdown
    - Functional Breakdown

- Common
- Physical Architecture
  - Physical Architecture Blank
    - [PAB] Physical System
  - Physical Components - Logical Components
    - New Physical Components - Logical Components
  - Physical Components/Actors - Physical Functions
    - New Physical Components/Actors - Physical Functions
  - Physical Data Flow Blank
    - [PDFB] Speed Sensing

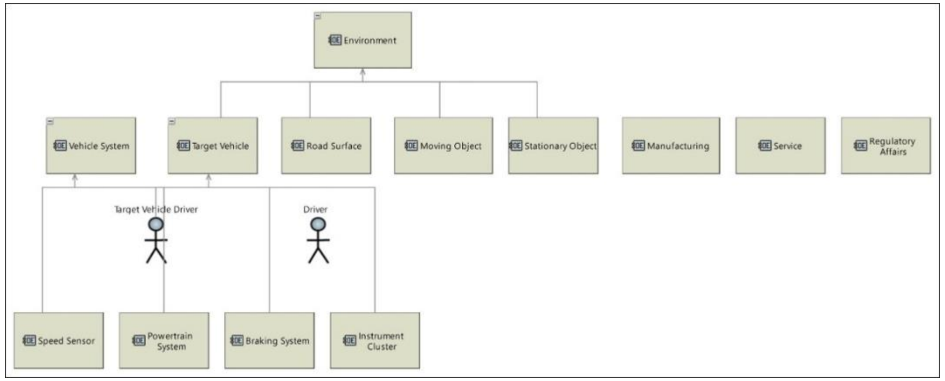
Architecture Development Pyramid: ARCADIA/Capella

Source: Shashank Alai IUPUI Systems Engineering MS thesis presentation (2019); used with permission. Access to thesis available upon request or at https://hdl.handle.net/1805/19766

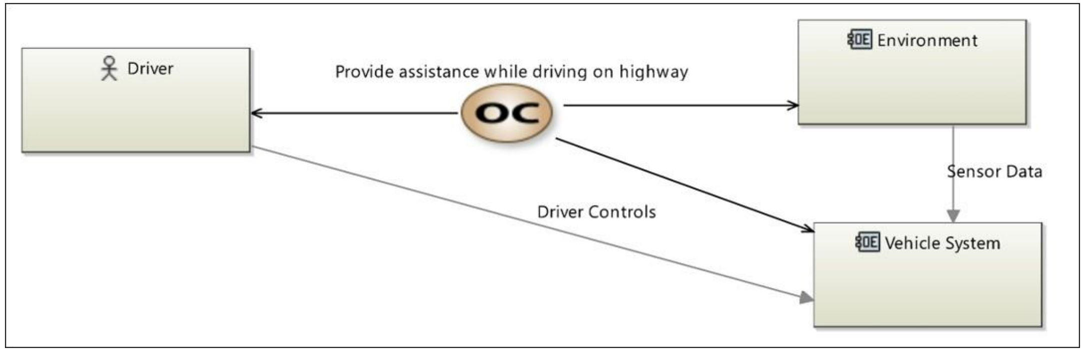
Operational (Needs) Analysis



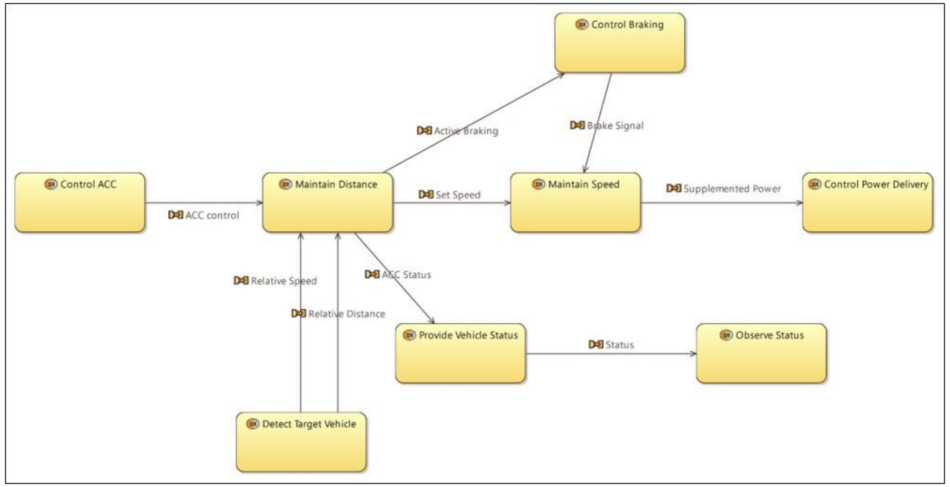
most abstract
Operational Analysis
System Requirements Analysis
Logical Architecture Definition
Physical Architecture Definition
least abstract



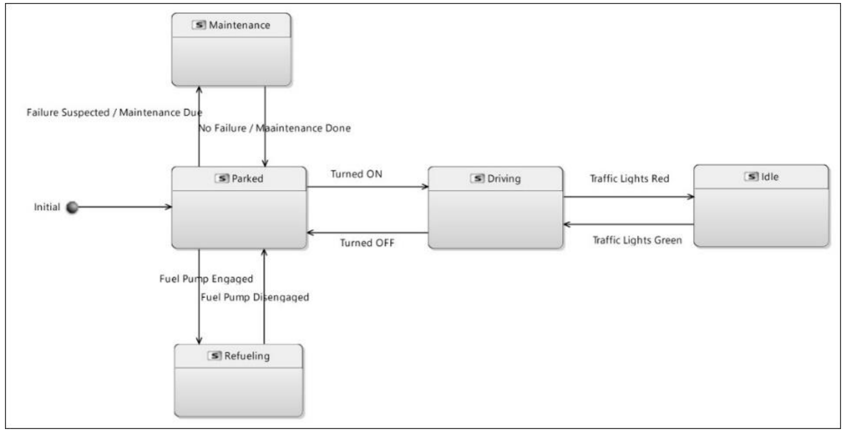
ACC System *Operational Entities Breakdown [OED]*



Capella operational capability using *Operational Capability Blank Diagram [OCB]*

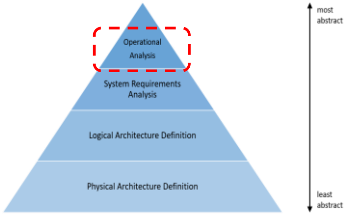


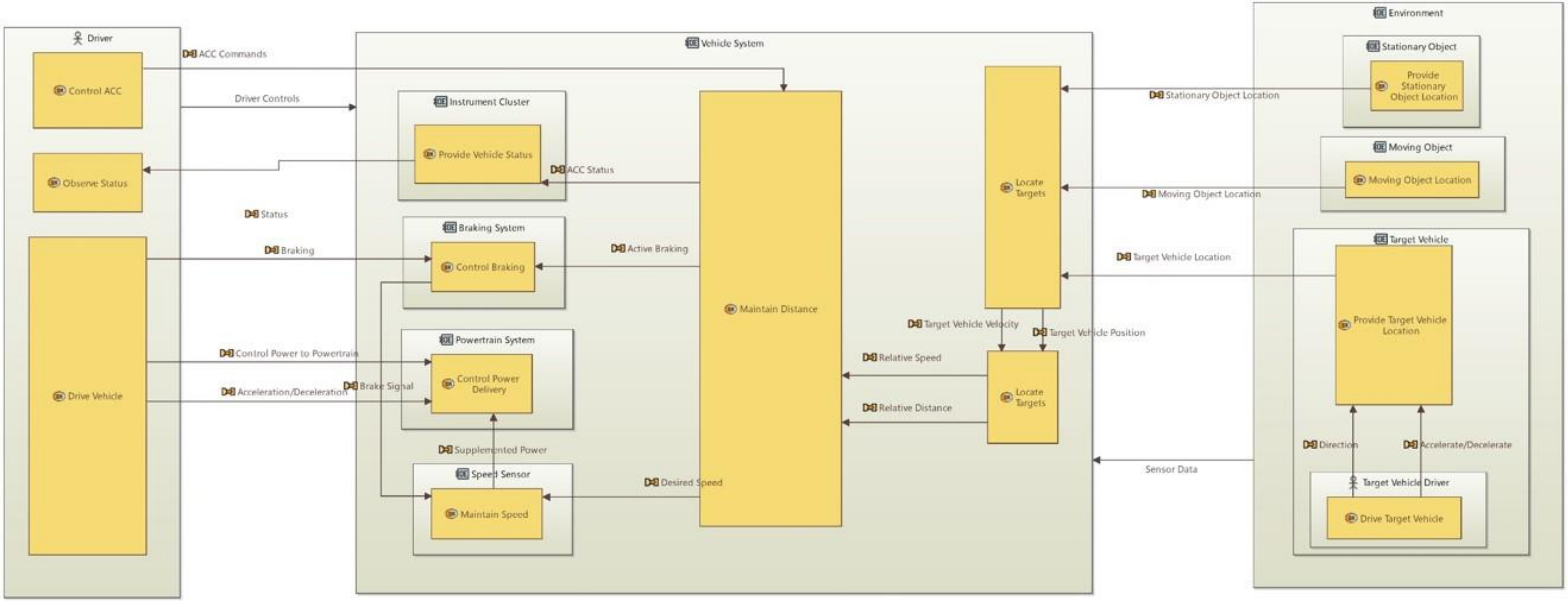
Provide ACC operation modeled using Operational Activity Interaction Blank [OAIB]



Vehicle Operational State using *Modes and States Machine Diagram [MSM]*

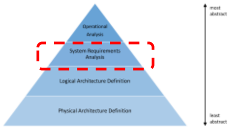
Operational Analysis, cont.

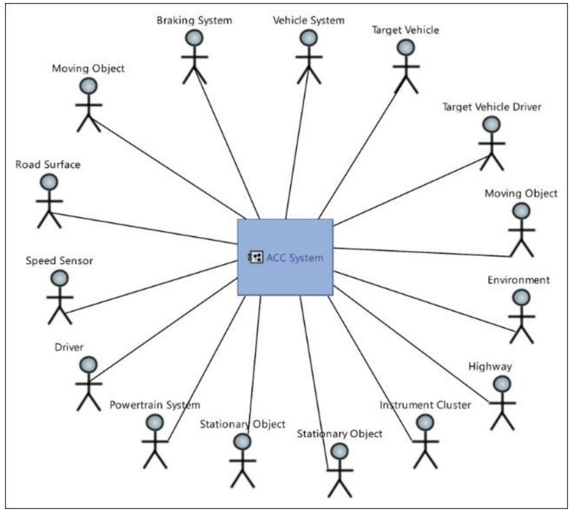

Operational Analysis
System Requirements Analysis
Logical Architecture Definition
Physical Architecture Definition
most abstract
least abstract



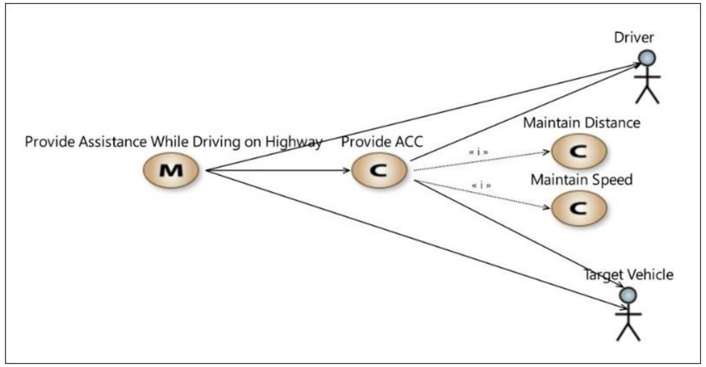
ACC *Operational Architecture Blank Diagram [OAB]*

System Requirements (Operational Needs) Analysis

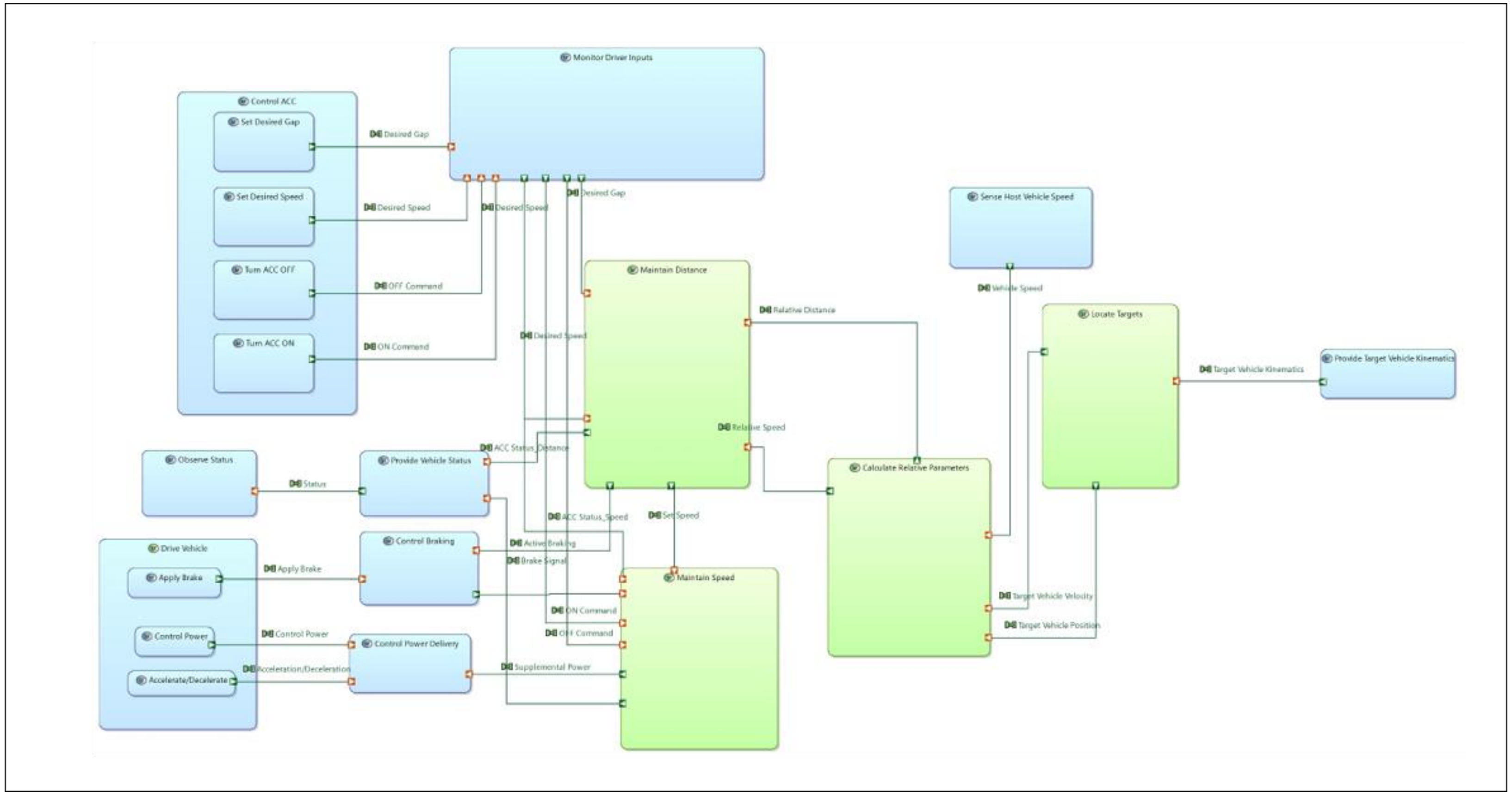

Operational Analysis
System Requirements Analysis
Logical Architecture Definition
Physical Architecture Definition
most abstract
least abstract



System Context using *Contextual System Actors [CSA]*

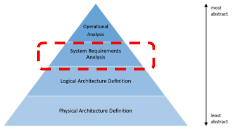


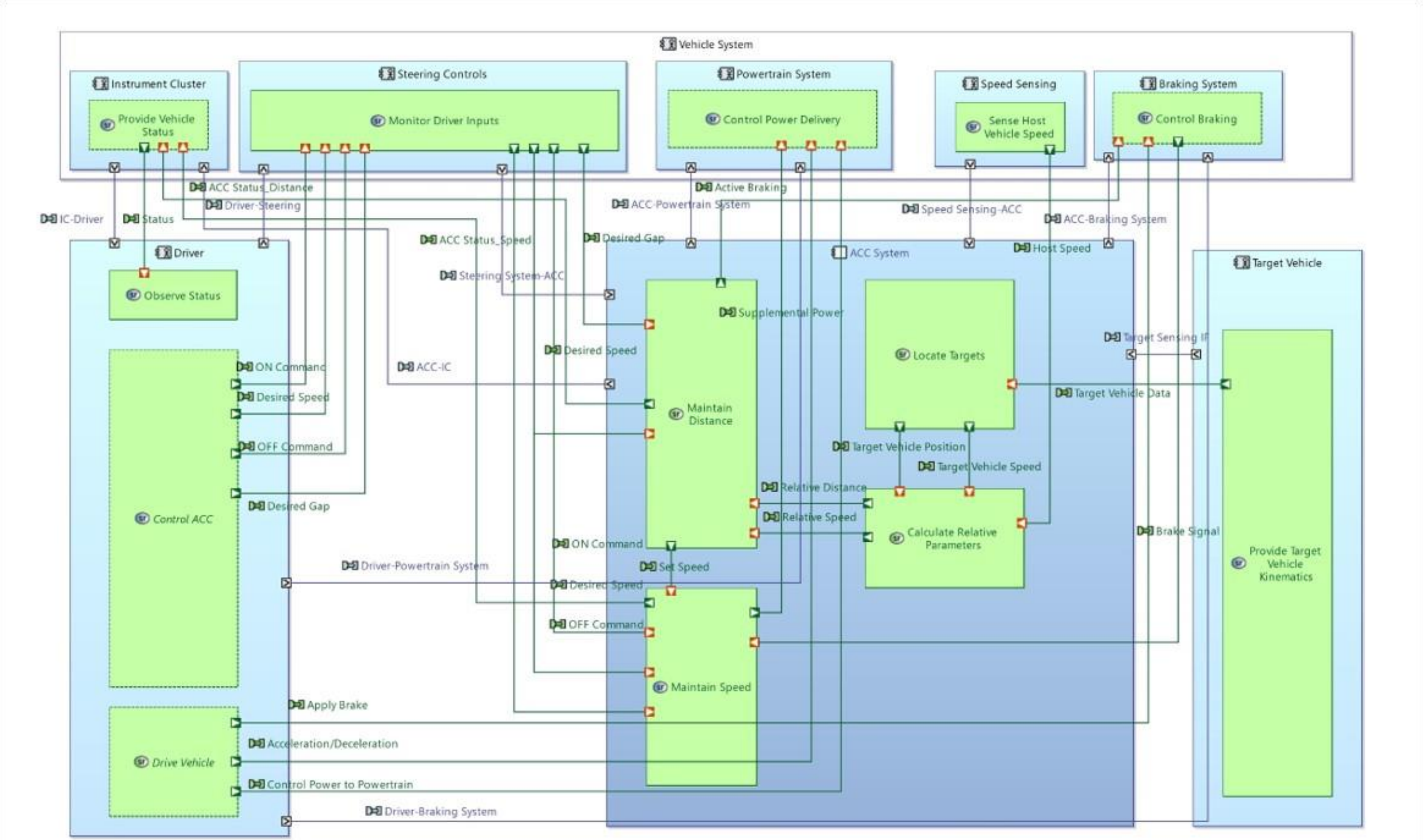
Capella system capability using *Missions Capabilities Blank Diagram [MCB]*



'Provide ACC' functions modeled using *System Data Flow Blank [SDFB]* diagram

System Requirements Analysis

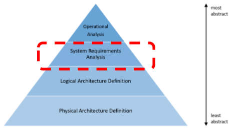

Operational Analysis
System Requirements Analysis
Logical Architecture Definition
Physical Architecture Definition
most abstract
least abstract

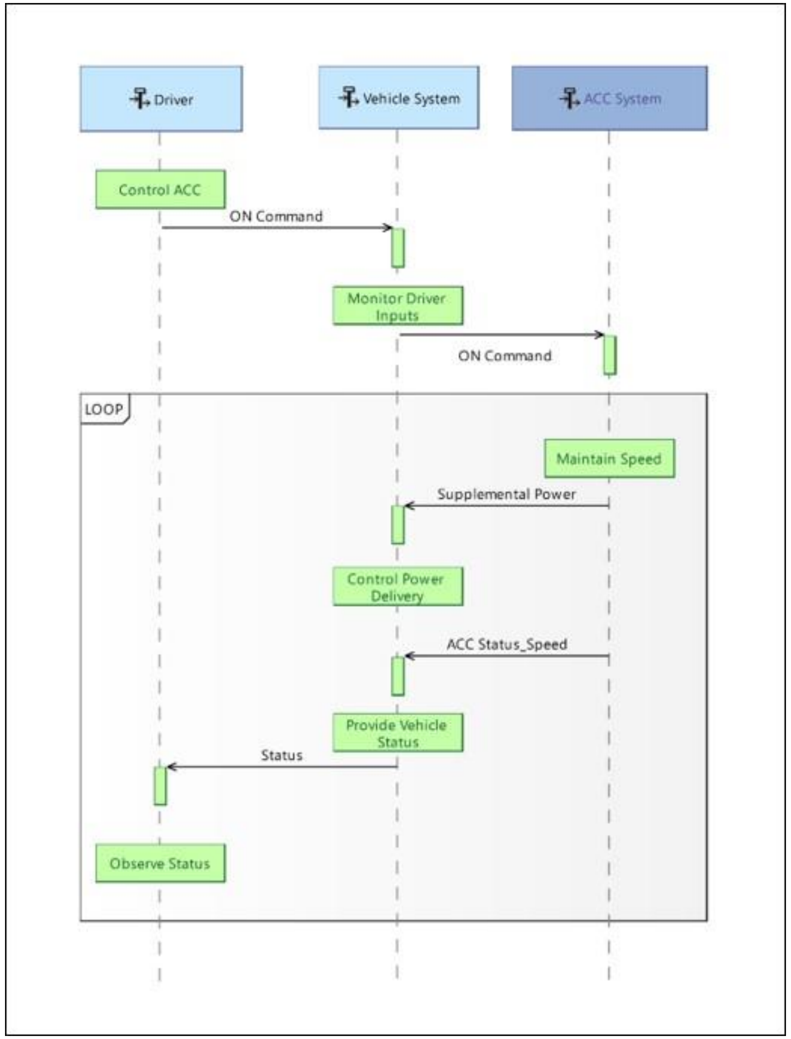


ACC system architecture using the *System Architecture Blank [SAB]* diagram

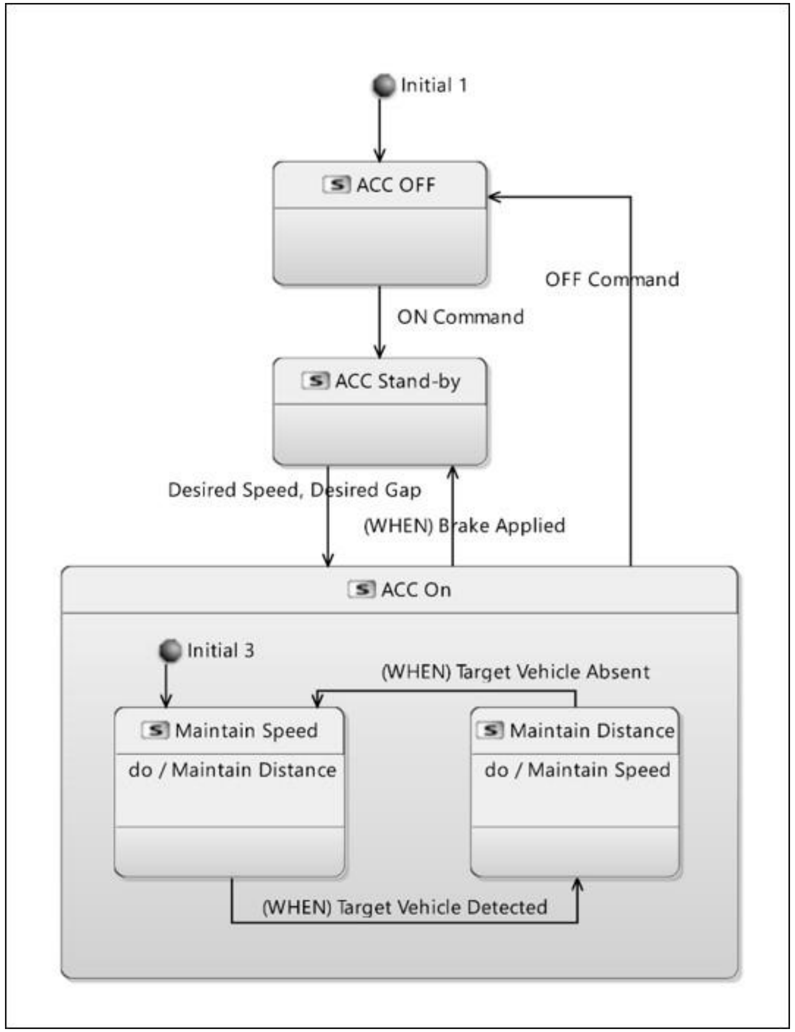
SIEMENS

System Requirements Analysis, cont.


Operational Analysis
System Requirements Analysis
Logical Architecture Definition
Physical Architecture Definition
most abstract
least abstract

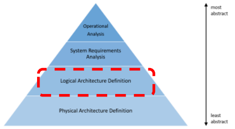


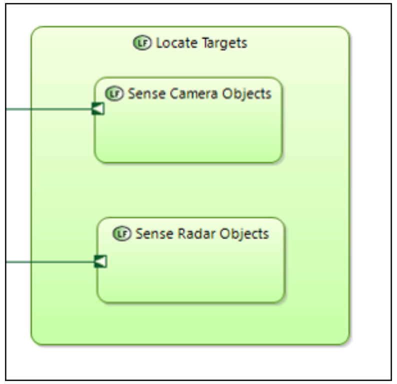
‘Turn ACC ON’ exchange scenario using the *Exchange Scenario [ES]* diagram



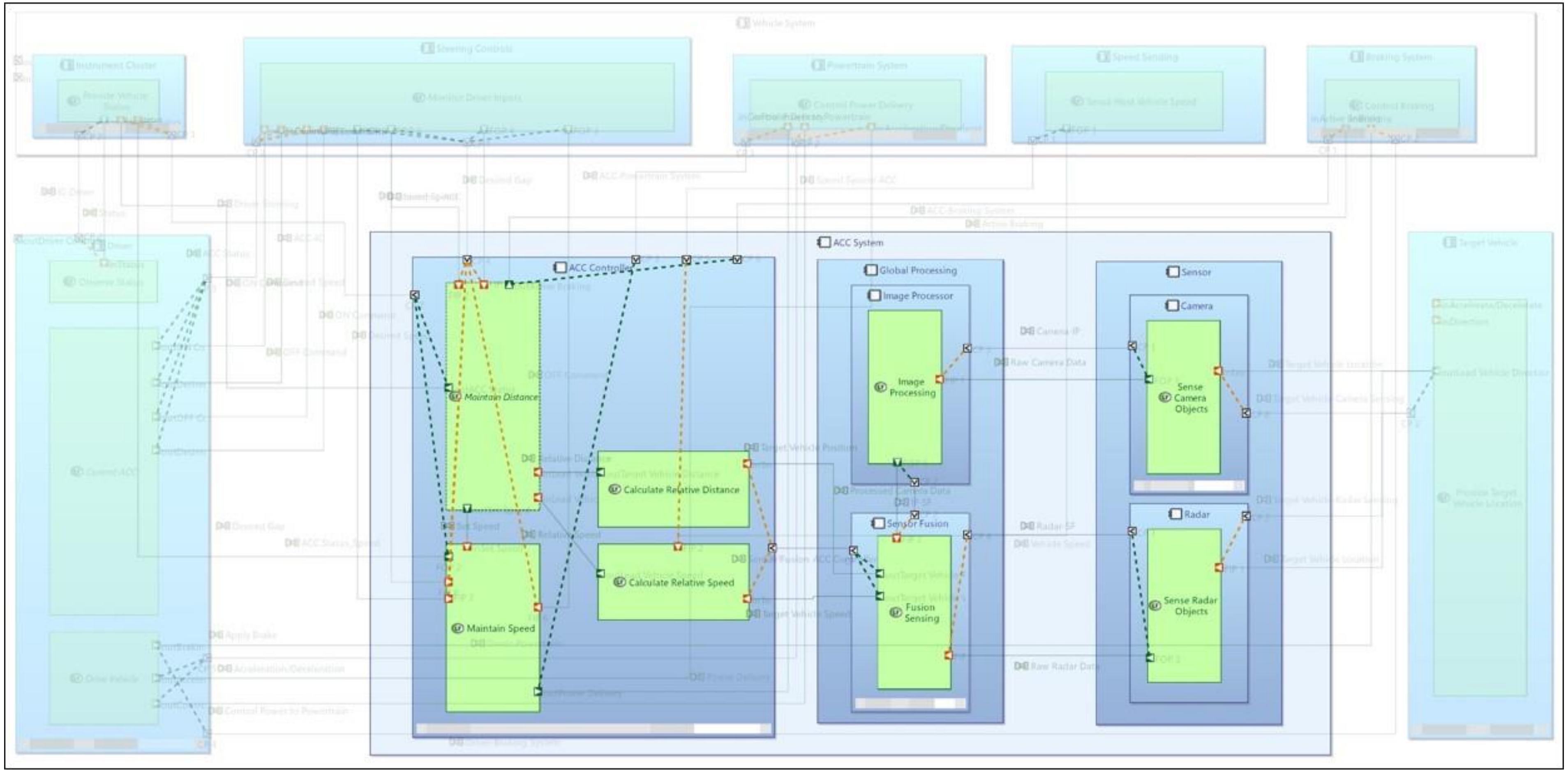
ACC System States using *Mode State Machine [MSM]* diagram

Logical Architecture Definition


Operational Analysis
System Requirements Analysis
Logical Architecture Definition
Physical Architecture Definition
most abstract
least abstract



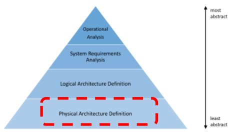
Logical function decomposition



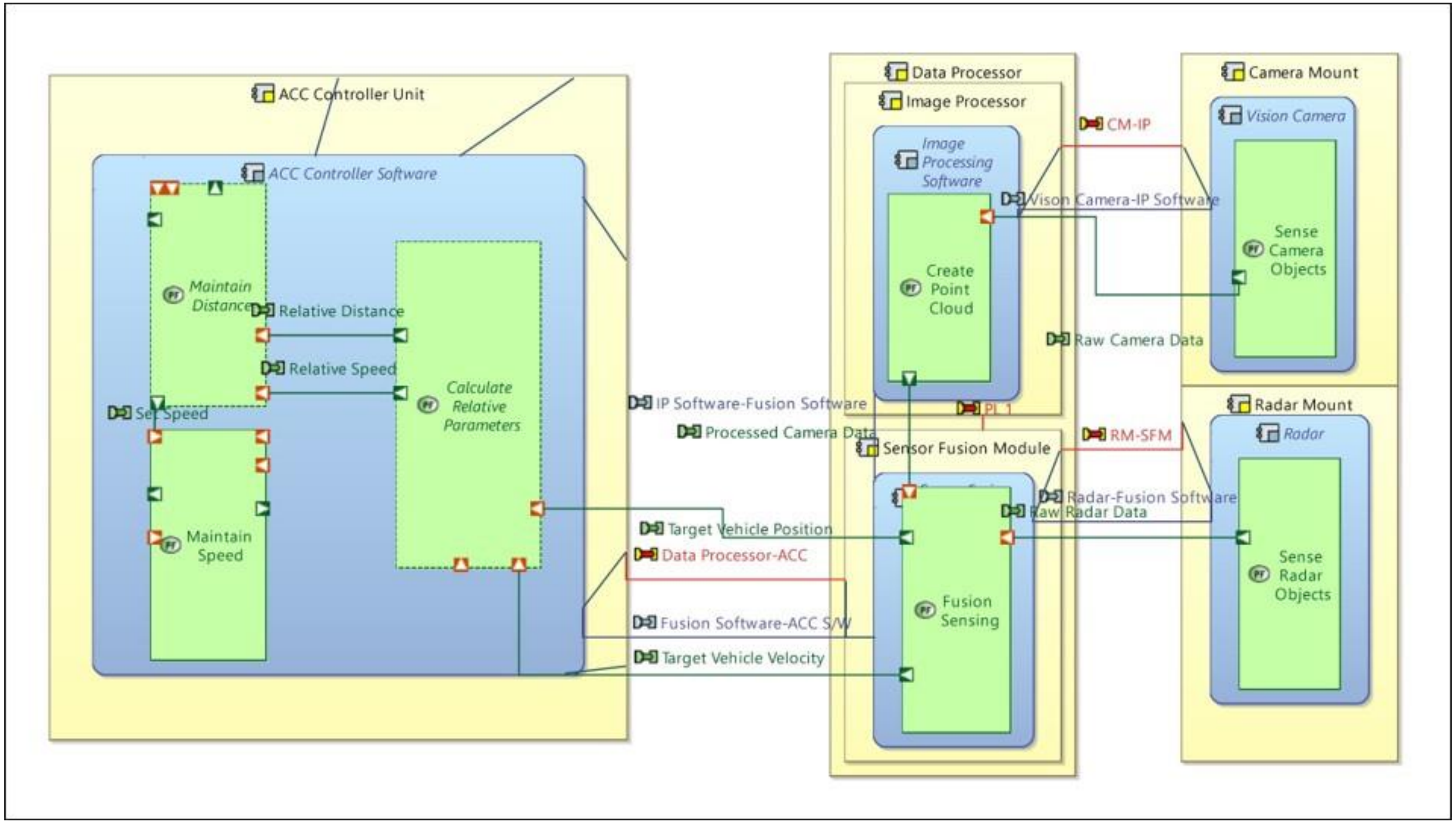
ACC System logical architecture modeled in *Logical Architecture Blank [LAB]* diagram

SIEMENS

Physical Architecture Definition

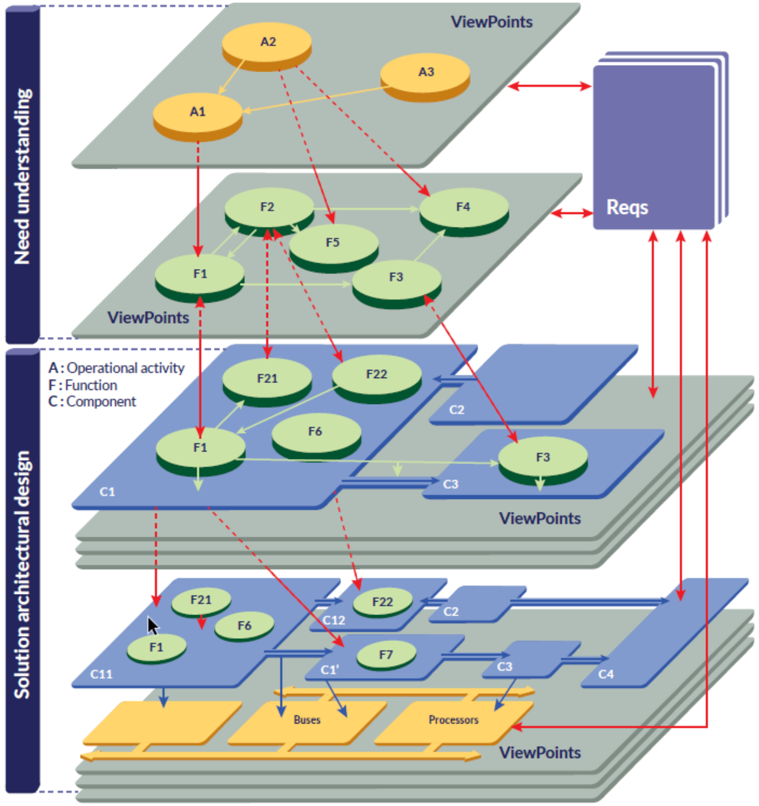


Operational Analysis
System Requirements Analysis
Logical Architecture Definition
Physical Architecture Definition
most abstract
least abstract



ACC Physical Architecture modeled in *Physical Architecture Blank [PAB]* diagram

Four Layers: Diagrams are representations of the same underlying model which knows the relationships

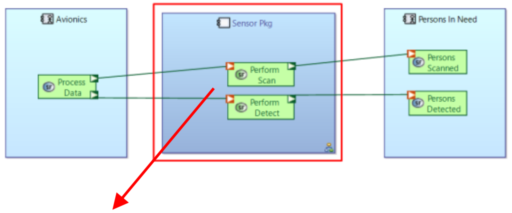


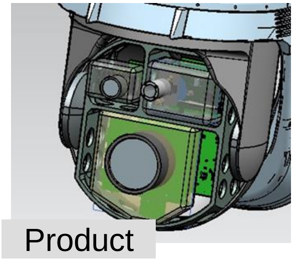
Operational Analysis
What the users of the system need to accomplish

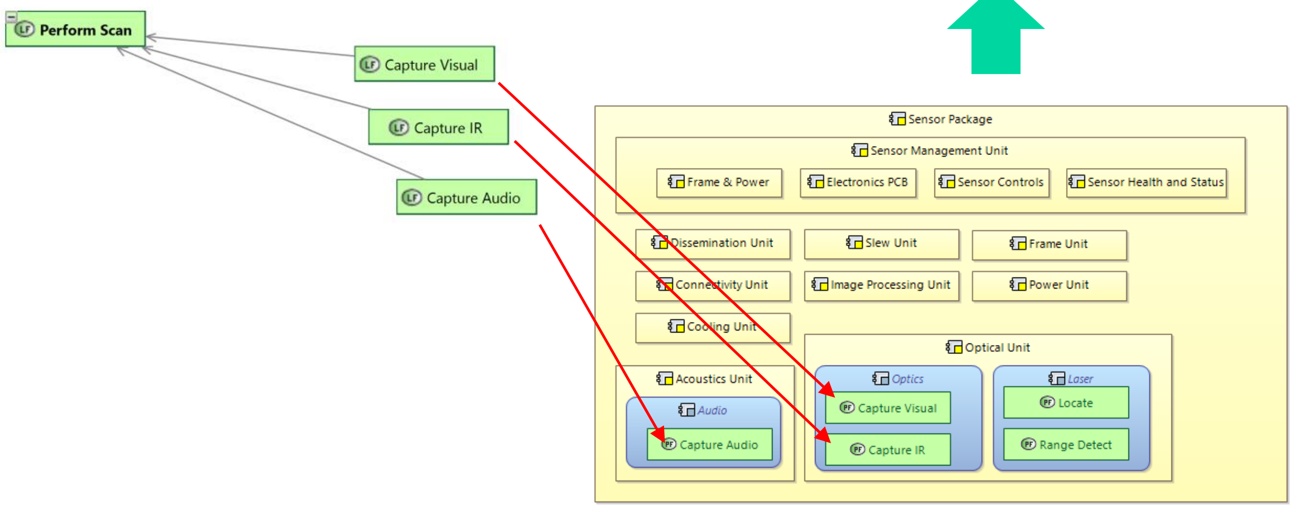
Functional & Non Functional Need
What the system has to accomplish for the users

Logical Architecture
How the system will work to fulfill expectations

Physical Architecture
How the system will be developed and built







SIEMENS

Agenda

Methodology-Driven MBSE

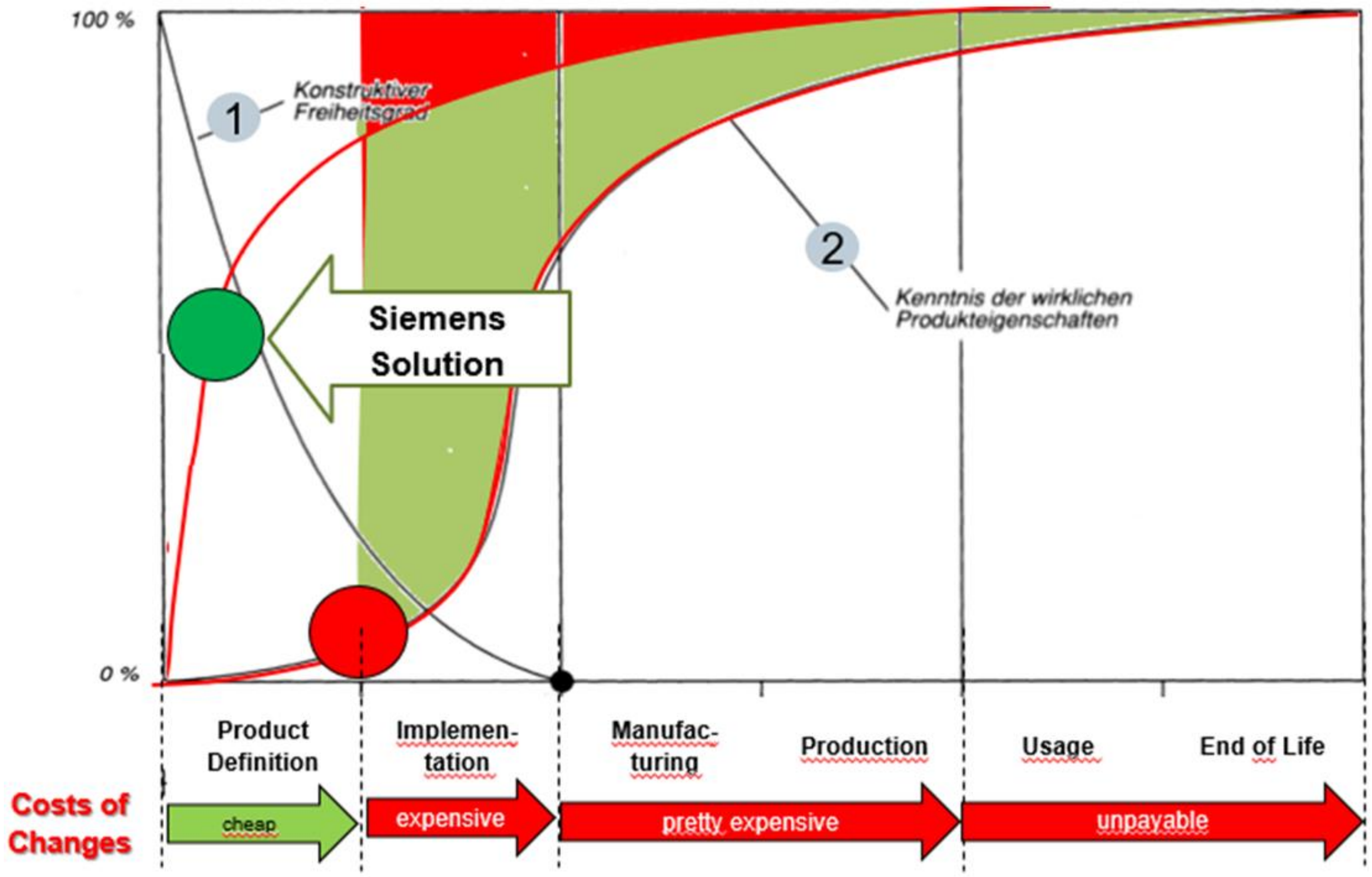
Model functional-system architecture

Advanced systems management with product architecture

Re-use product architecture through multi-domain viewpoints

Close-the-loop with the product architecture

Benefits of Analyzing and Architecting Functional Systems

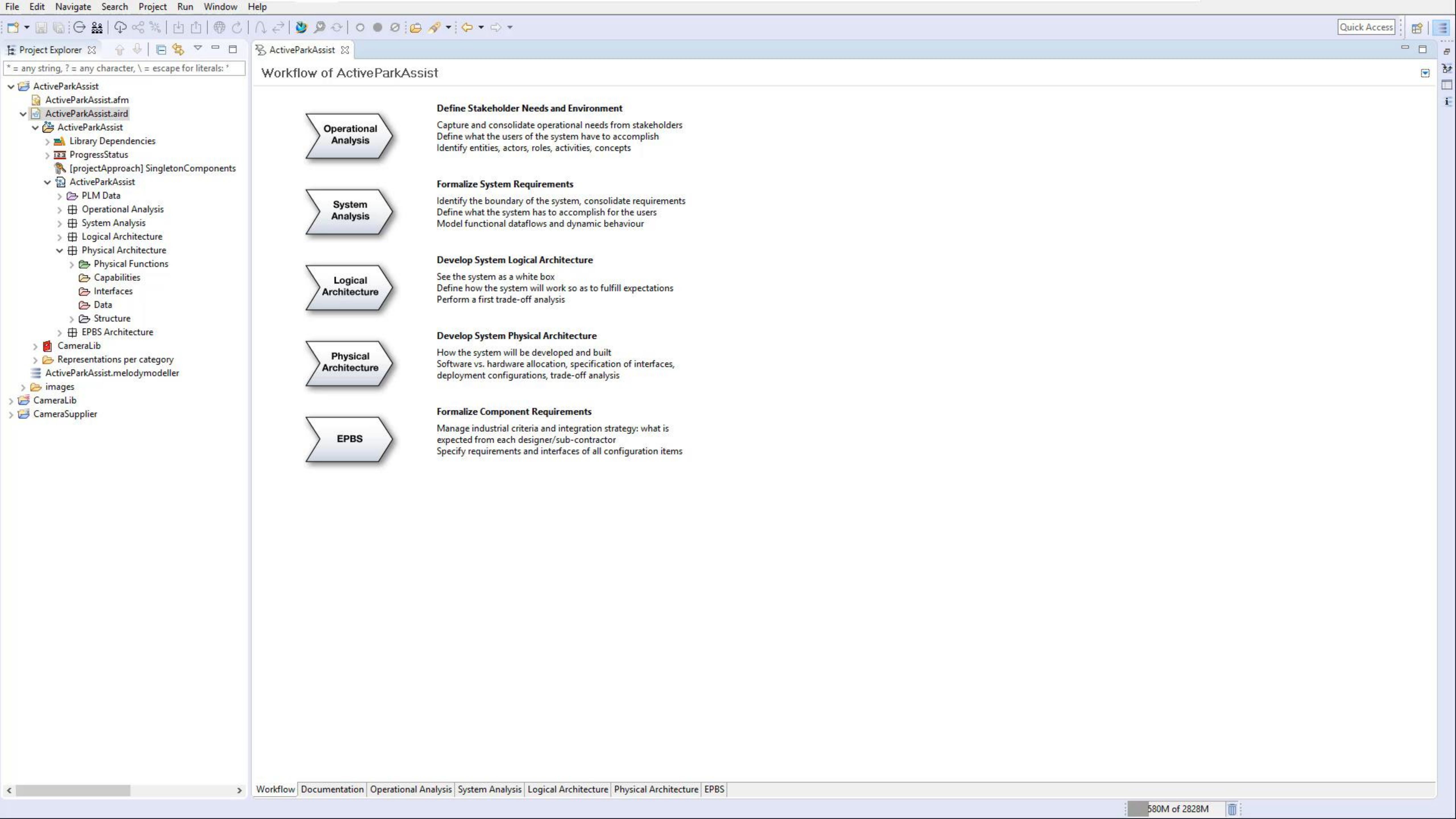


70% of product, production and operating costs **are determined in the early phase** of the product's definition

1 Degree of Constructive Freedom

2 Knowledge about the real product properties and behavior

Quelle: www.daswirtschaftslexikon.com
PEP = Produktentwicklungsprozess



Workflow of ActiveParkAssist

Define Stakeholder Needs and Environment

Capture and consolidate operational needs from stakeholders
Define what the users of the system have to accomplish
Identify entities, actors, roles, activities, concepts

Formalize System Requirements

Identify the boundary of the system, consolidate requirements
Define what the system has to accomplish for the users
Model functional dataflows and dynamic behaviour

Develop System Logical Architecture

See the system as a white box
Define how the system will work so as to fulfill expectations
Perform a first trade-off analysis

Develop System Physical Architecture

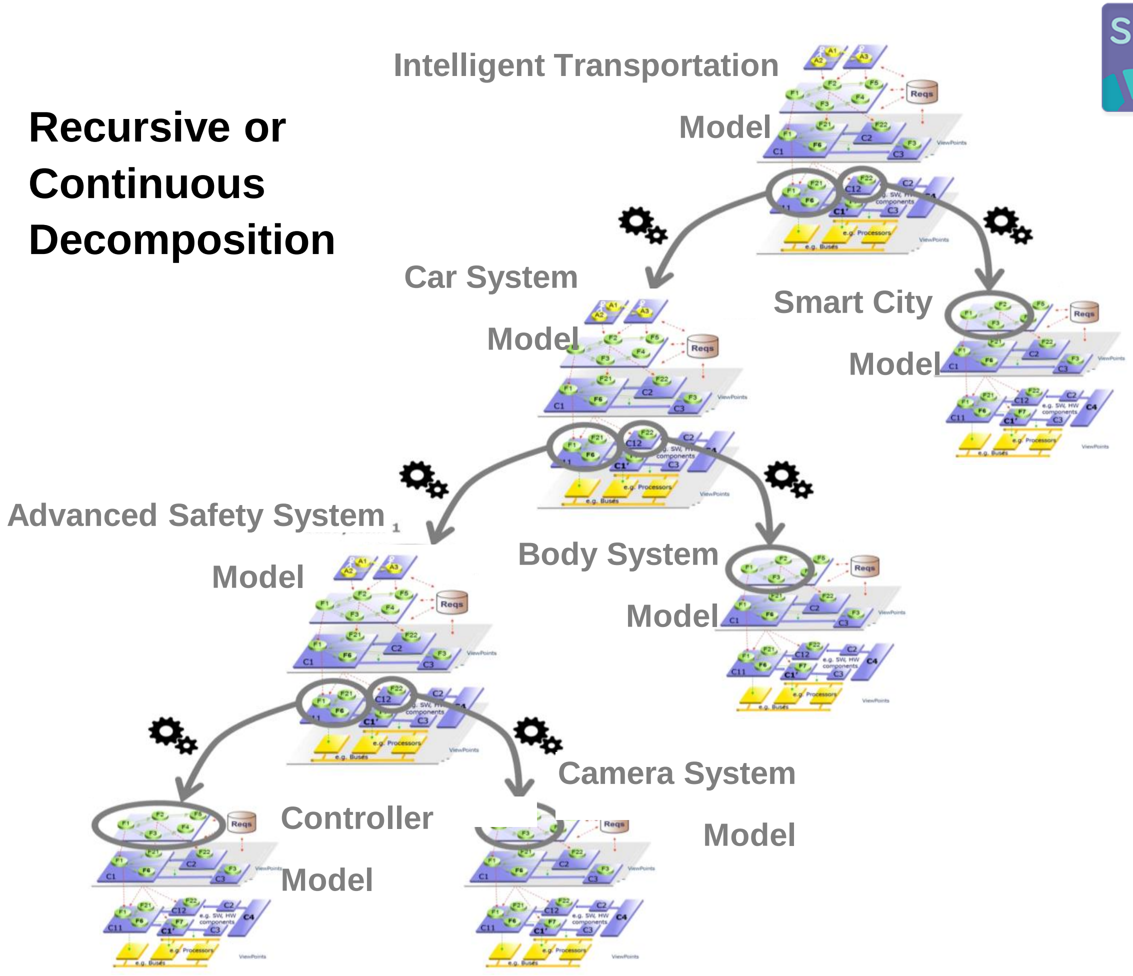
How the system will be developed and built
Software vs. hardware allocation, specification of interfaces,
deployment configurations, trade-off analysis

Formalize Component Requirements

Manage industrial criteria and integration strategy: what is
expected from each designer/sub-contractor
Specify requirements and interfaces of all configuration items

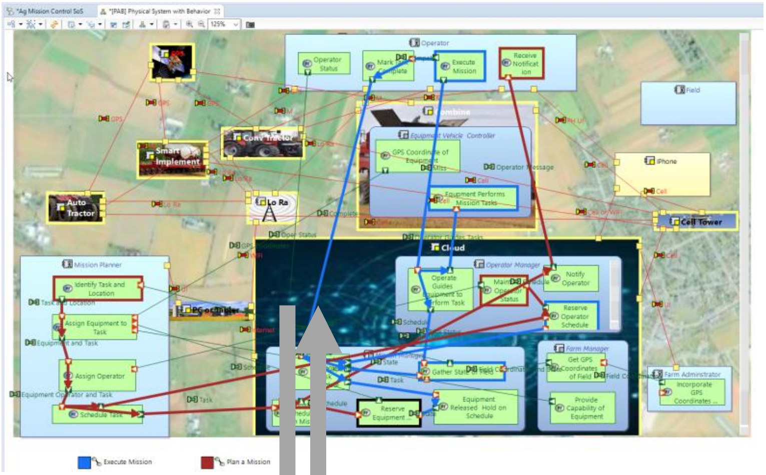
System to Subsystem Transition Supports Rich SI-Supplier Collaboration

Recursive or Continuous Decomposition



System Modeling Workbench

Integrator System of Systems Model

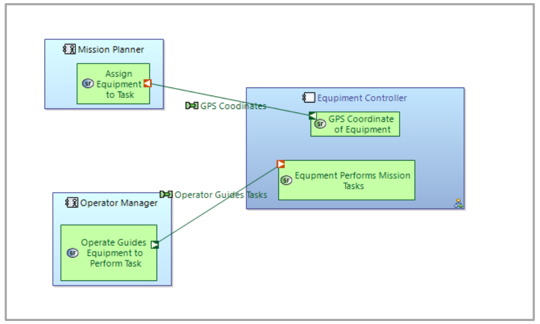


Hand-off

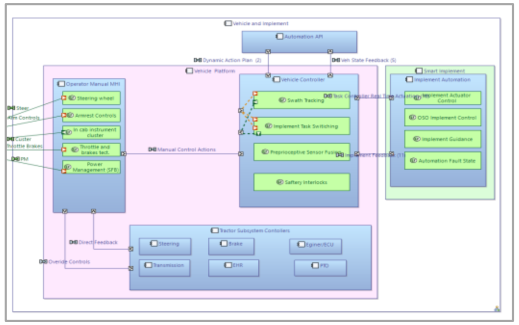
Only share limited IP

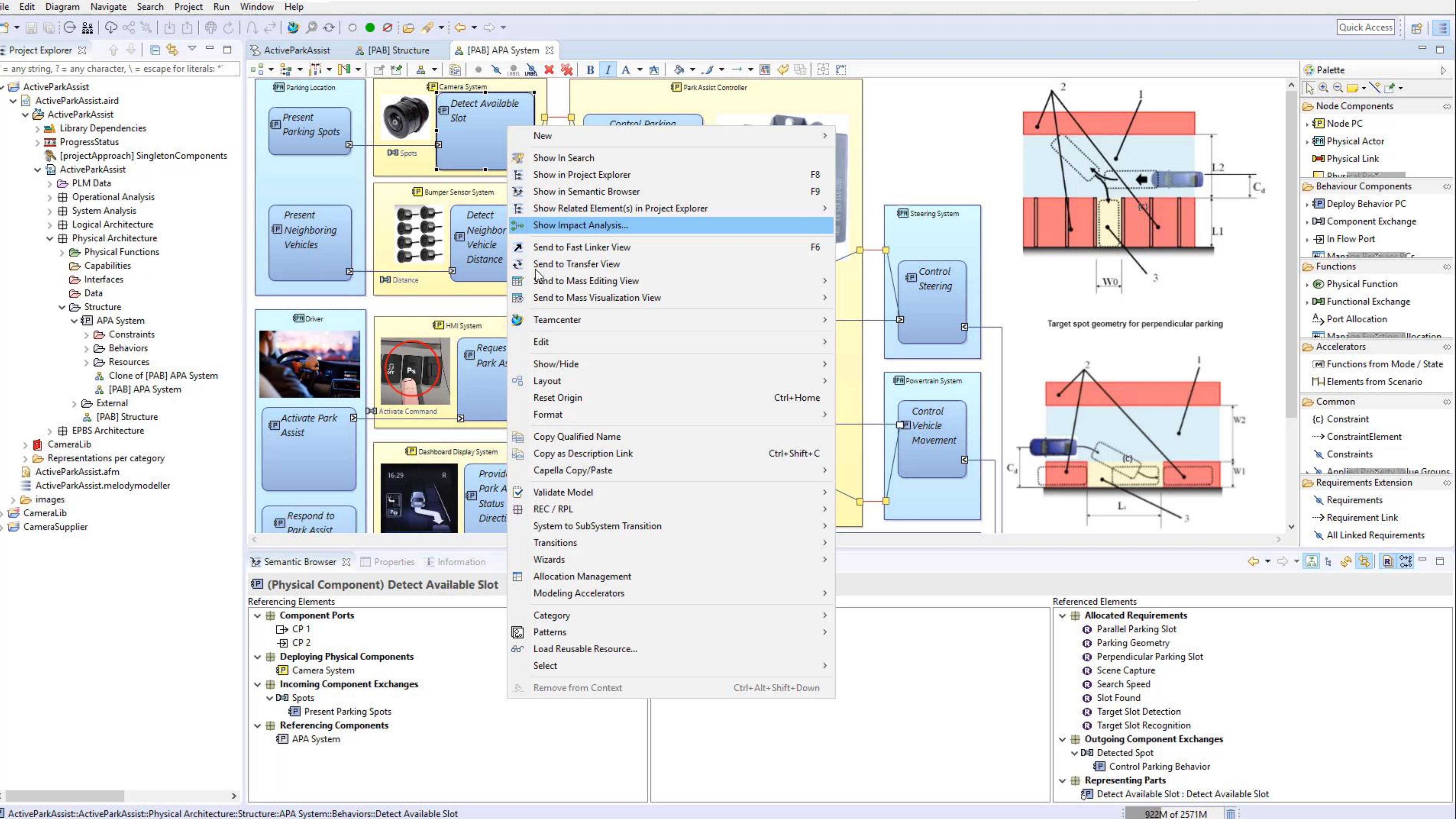
Integrate

Supplier System Model



Supplier Enriched Library Component




Edit Diagram Navigate Search Project Run Window Help
Quick Access
Project Explorer
= any string, ? = any character, \ = escape for literals: *'
ActivePark Assist
ActiveParkAssist.aird
ActiveParkAssist
Library Dependencies
ProgressStatus
[projectApproach] SingletonComponents
ActiveParkAssist
PLM Data
Operational Analysis
System Analysis
Logical Architecture
Physical Architecture
Physical Functions
Capabilities
Interfaces
Data
Structure
APA System
Constraints
Behaviors
Resources
Clone of [PAB] APA System
[PAB] APA System
External
[PAB] Structure
EPBS Architecture
CameraLib
Representations per category
ActiveParkAssist.afm
ActiveParkAssist.melodymodeller
images
CameraLib
CameraSupplier
ActiveParkAssist
[PAB] Structure
[PAB] APA System
Parking Location
Present Parking Spots
Present Neighboring Vehicles
Camera System
Detect Available Slot
Spots
Bumper Sensor System
Detect Neighbor Vehicle Distance
Distance
Park Assist Controller
Control Parking
Driver
Activate Park Assist
Respond to Park Assist
HMI System
Request Park A
Activate Command
Dashboard Display System
Provide Park A Status Directi
Steering System
Control Steering
Powertrain System
Control Vehicle Movement
Target spot geometry for perpendicular parking
New
Show In Search
Show in Project Explorer F8
Show in Semantic Browser F9
Show Related Element(s) in Project Explorer
Show Impact Analysis...
Send to Fast Linker View F6
Send to Transfer View
Send to Mass Editing View
Send to Mass Visualization View
Teamcenter
Edit
Show/Hide
Layout
Reset Origin Ctrl+Home
Format
Copy Qualified Name
Copy as Description Link Ctrl+Shift+C
Capella Copy/Paste
Validate Model
REC / RPL
System to SubSystem Transition
Transitions
Wizards
Allocation Management
Modeling Accelerators
Category
Patterns
Load Reusable Resource...
Select
Remove from Context Ctrl+Alt+Shift+Down
Palette
Node Components
Node PC
Physical Actor
Physical Link
Behaviour Components
Deploy Behavior PC
Component Exchange
In Flow Port
Functions
Physical Function
Functional Exchange
Port Allocation
Accelerators
Functions from Mode / State
Elements from Scenario
Common
Constraint
ConstraintElement
Constraints
Requirements Extension
Requirements
Requirement Link
All Linked Requirements
Semantic Browser
Properties
Information
(Physical Component) Detect Available Slot
Referencing Elements
Component Ports
CP 1
CP 2
Deploying Physical Components
Camera System
Incoming Component Exchanges
Spots
Present Parking Spots
Referencing Components
APA System
Referenced Elements
Allocated Requirements
Parallel Parking Slot
Parking Geometry
Perpendicular Parking Slot
Scene Capture
Search Speed
Slot Found
Target Slot Detection
Target Slot Recognition
Outgoing Component Exchanges
Detected Spot
Control Parking Behavior
Representing Parts
Detect Available Slot : Detect Available Slot
ActiveParkAssist::ActiveParkAssist::Physical Architecture::Structure::APA System::Behaviors::Detect Available Slot
922M of 2571M

Agenda

Model functional-system architecture

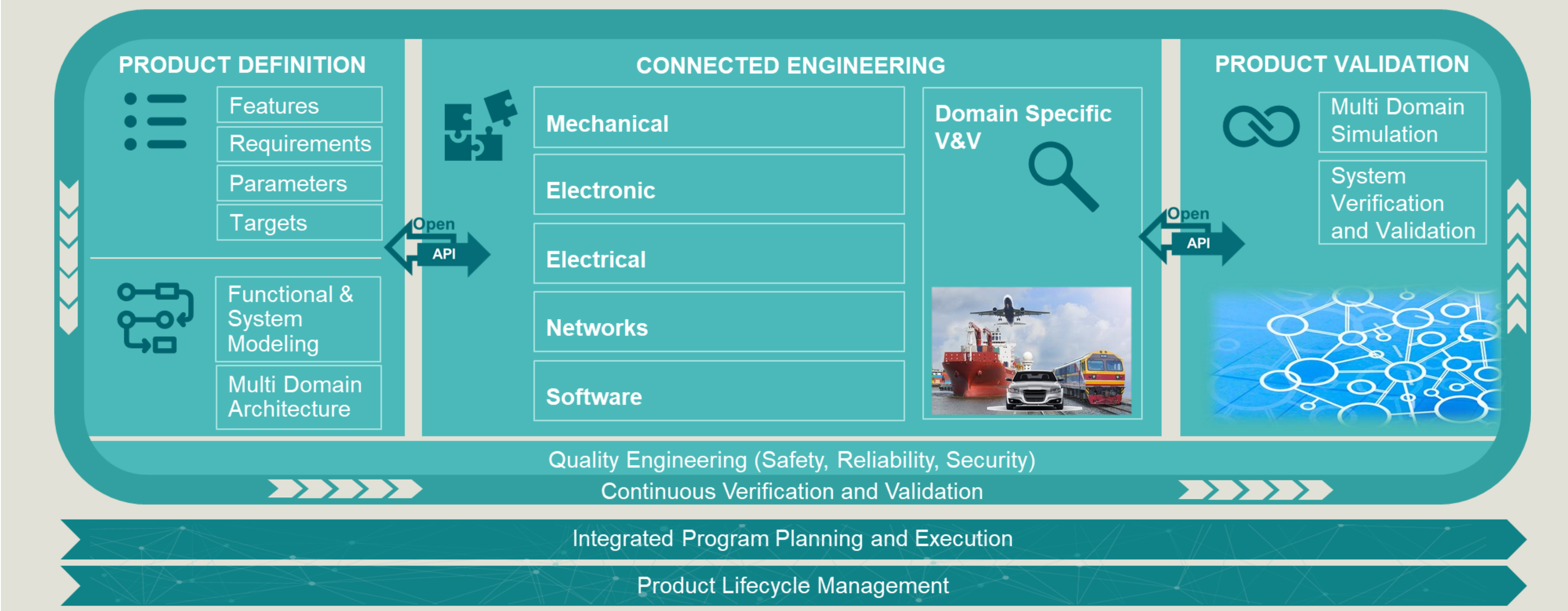
Advanced systems management with product architecture

Re-use product architecture through multi-domain viewpoints

Close-the-loop with the product architecture

Model-Based Systems Engineering Writ Large

Start Integrated, Stay Integrated



PRODUCT DEFINITION

- Features
- Requirements
- Parameters
- Targets
- Functional & System Modeling
- Multi Domain Architecture

Open API

CONNECTED ENGINEERING

- Mechanical
- Electronic
- Electrical
- Networks
- Software

Domain Specific V&V

Open API

PRODUCT VALIDATION

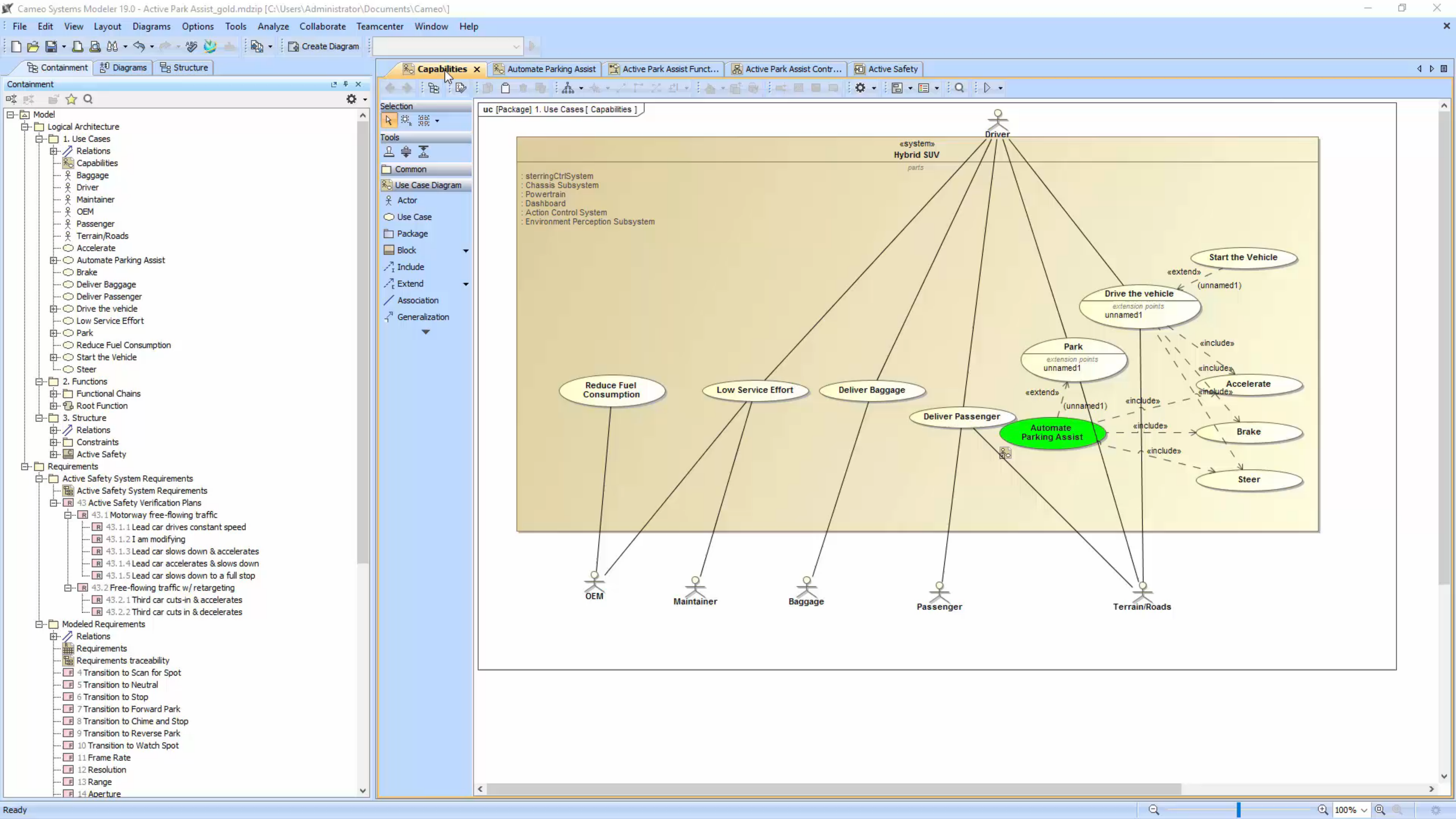
- Multi Domain Simulation
- System Verification and Validation

Quality Engineering (Safety, Reliability, Security)

Continuous Verification and Validation

Integrated Program Planning and Execution

Product Lifecycle Management

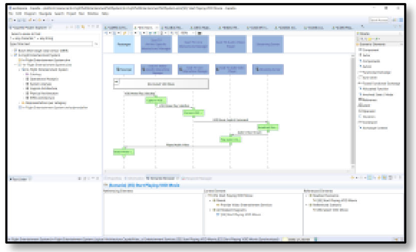

Cameo Systems Modeler 19.0 - Active Park Assist_gold.mdzip [C:\Users\Administrator\Documents\Cameo\]
File
Edit
View
Layout
Diagrams
Options
Tools
Analyze
Collaborate
Teamcenter
Window
Help
Create Diagram
Containment
Diagrams
Structure
Capabilities
Automate Parking Assist
Active Park Assist Funct...
Active Park Assist Contr...
Active Safety
Model
Logical Architecture
1. Use Cases
Relations
Capabilities
Baggage
Driver
Maintainer
OEM
Passenger
Terrain/Roads
Accelerate
Automate Parking Assist
Brake
Deliver Baggage
Deliver Passenger
Drive the vehicle
Low Service Effort
Park
Reduce Fuel Consumption
Start the Vehicle
Steer
2. Functions
Functional Chains
Root Function
3. Structure
Relations
Constraints
Active Safety
Requirements
Active Safety System Requirements
Active Safety System Requirements
43 Active Safety Verification Plans
43.1 Motorway free-flowing traffic
43.1.1 Lead car drives constant speed
43.1.2 I am modifying
43.1.3 Lead car slows down & accelerates
43.1.4 Lead car accelerates & slows down
43.1.5 Lead car slows down to a full stop
43.2 Free-flowing traffic w/ retargeting
43.2.1 Third car cuts-in & accelerates
43.2.2 Third car cuts in & decelerates
Modeled Requirements
Relations
Requirements
Requirements traceability
4 Transition to Scan for Spot
5 Transition to Neutral
6 Transition to Stop
7 Transition to Forward Park
8 Transition to Chime and Stop
9 Transition to Reverse Park
10 Transition to Watch Spot
11 Frame Rate
12 Resolution
13 Range
14 Aperture
Selection
Tools
Common
Use Case Diagram
Actor
Use Case
Package
Block
Include
Extend
Association
Generalization
uc [Package] 1. Use Cases [Capabilities]
Driver
«system»
Hybrid SUV
parts
: sterringCtrlSystem
: Chassis Subsystem
: Powertrain
: Dashboard
: Action Control System
: Environment Perception Subsystem
Start the Vehicle
«extend»
(unnamed1)
Drive the vehicle
extension points
unnamed1
«include»
Park
extension points
unnamed1
«include»
Accelerate
«include»
«extend»
(unnamed1)
«include»
Reduce Fuel Consumption
Low Service Effort
Deliver Baggage
Deliver Passenger
Automate Parking Assist
«include»
Brake
«include»
Steer
OEM
Maintainer
Baggage
Passenger
Terrain/Roads
Ready
100%

Extend Your System Modeling Value (Forward Compatibility)

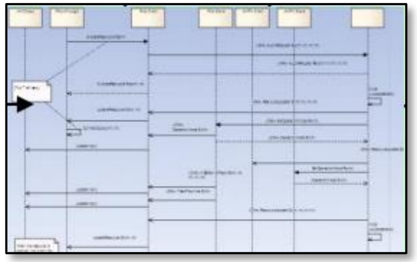
An open integration framework enabling interoperability among engineering models and tools in a diverse MBSE eco-system

- Enterprise system model lifecycle management, workflow, and collaboration where models are reusable and in Teamcenter
- Inform on dependencies with bidirectional traceability of RFLP data - visible in Teamcenter through Relationship Browser
- Manage change across multi-domain modeling
- Visualize and make decisions based on modeling data in dynamic views with Active Workspace
- Exchange and close-loop with multi-domain engineering for design, analysis, optimization and trade-off

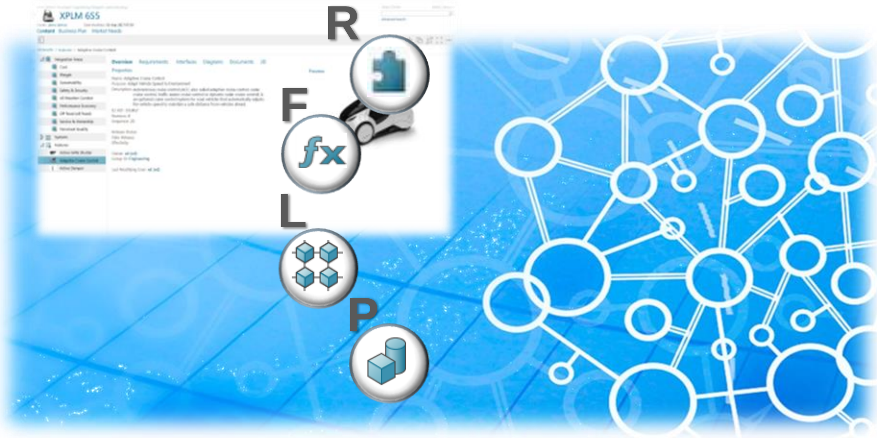
System Modeling Workbench (with Capella)

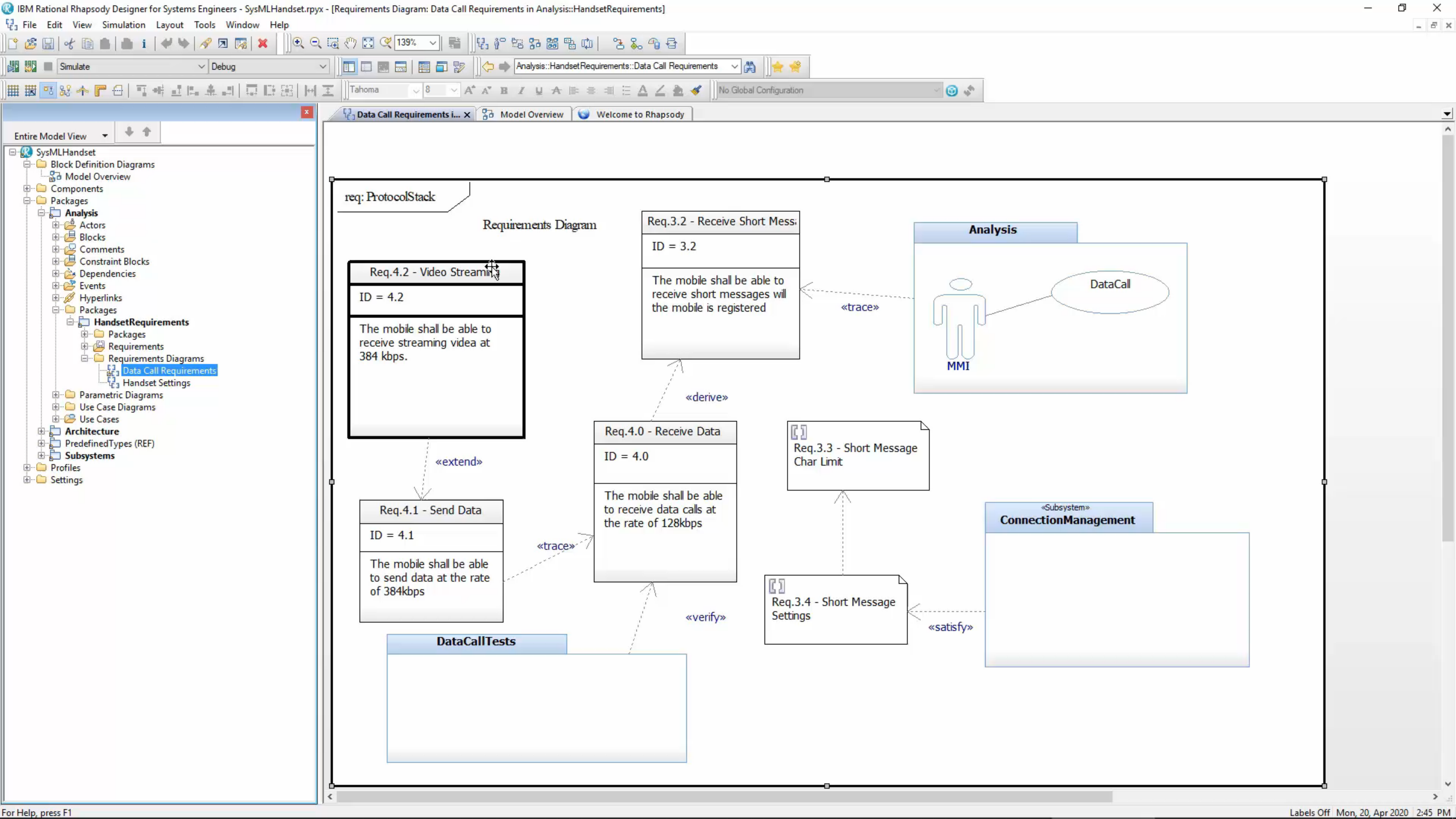


3rd Parties (Cameo, Rhapsody)



TEAMCENTER





IBM Rational Rhapsody Designer for Systems Engineers - SysMLHandset.rpyx - [Requirements Diagram: Data Call Requirements in Analysis::HandsetRequirements]

Req.3.2 - Receive Short Mess: The mobile shall be able to receive short messages will the mobile is registered

Req.4.2 - Video Streaming: The mobile shall be able to receive streaming videa at 384 kbps.

Req.4.0 - Receive Data: The mobile shall be able to receive data calls at the rate of 128kbps

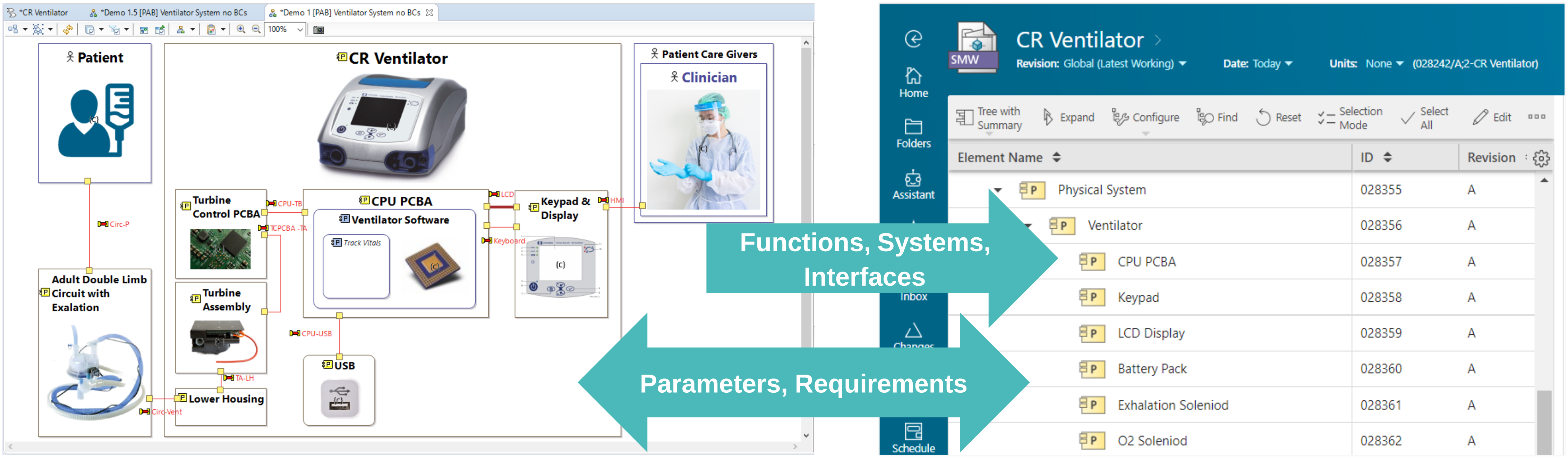
Req.4.1 - Send Data: The mobile shall be able to send data at the rate of 384kbps

Multi-Domain Product Architecture

Most Complete Backbone For Product Architecture

System Model

Digital Thread



CR Ventilator

Revision: Global (Latest Working) Date: Today Units: None (028242/A;2-CR Ventilator)

Element Name	ID	Revision
Physical System	028355	A
Ventilator	028356	A
CPU PCBA	028357	A
Keypad	028358	A
LCD Display	028359	A
Battery Pack	028360	A
Exhalation Soleniod	028361	A
O2 Soleniod	028362	A

Functions, Systems, Interfaces

Parameters, Requirements

Any Tool

Teamcenter

Agenda

Methodology-Driven MBSE

Model functional-system architecture

Advanced systems management with product architecture

Re-use product architecture through multi-domain viewpoints

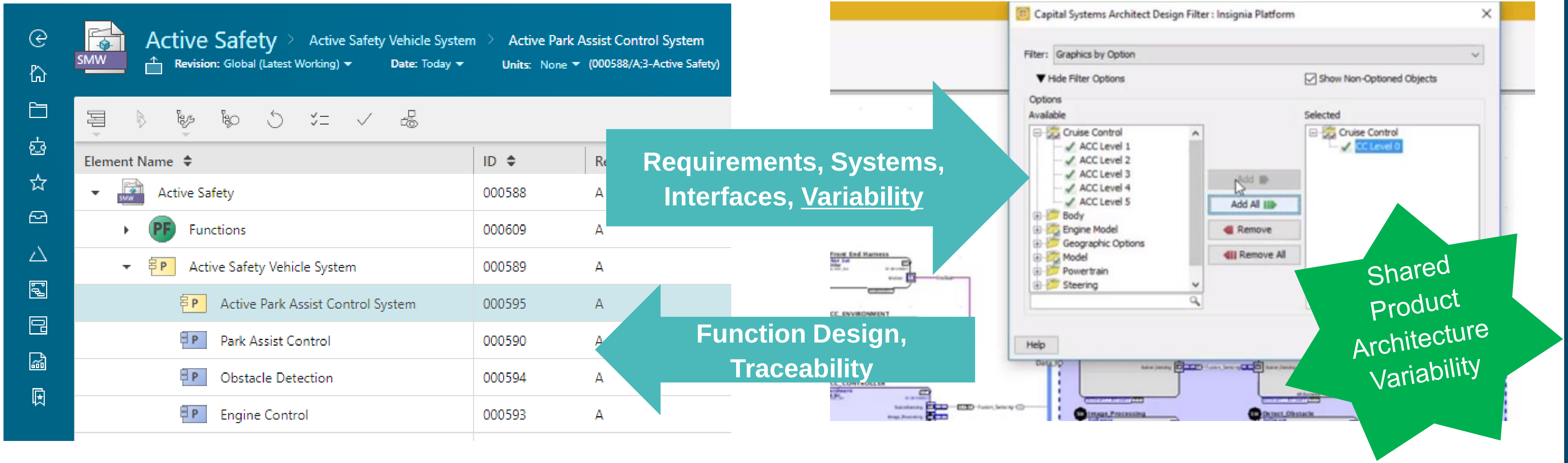
Close-the-loop with the product architecture

Connected Electrical Design

Generate configurable function design that is consistent

Product Architecture

Function Design



Active Safety > Active Safety Vehicle System > Active Park Assist Control System

Revision: Global (Latest Working) Date: Today Units: None (000588/A;3-Active Safety)

Element Name	ID	Re
Active Safety	000588	A
Functions	000609	A
Active Safety Vehicle System	000589	A
Active Park Assist Control System	000595	A
Park Assist Control	000590	A
Obstacle Detection	000594	A
Engine Control	000593	A

Capital Systems Architect Design Filter : Insignia Platform

Filter: Graphics by Option

▼ Hide Filter Options

☑ Show Non-Optioned Objects

Options

Available
- Cruise Control
  - ✓ ACC Level 1
  - ✓ ACC Level 2
  - ✓ ACC Level 3
  - ✓ ACC Level 4
  - ✓ ACC Level 5
- Body
- Engine Model
- Geographic Options
- Model
- Powertrain
- Steering

Selected
- Cruise Control
  - ✓ CC Level 0

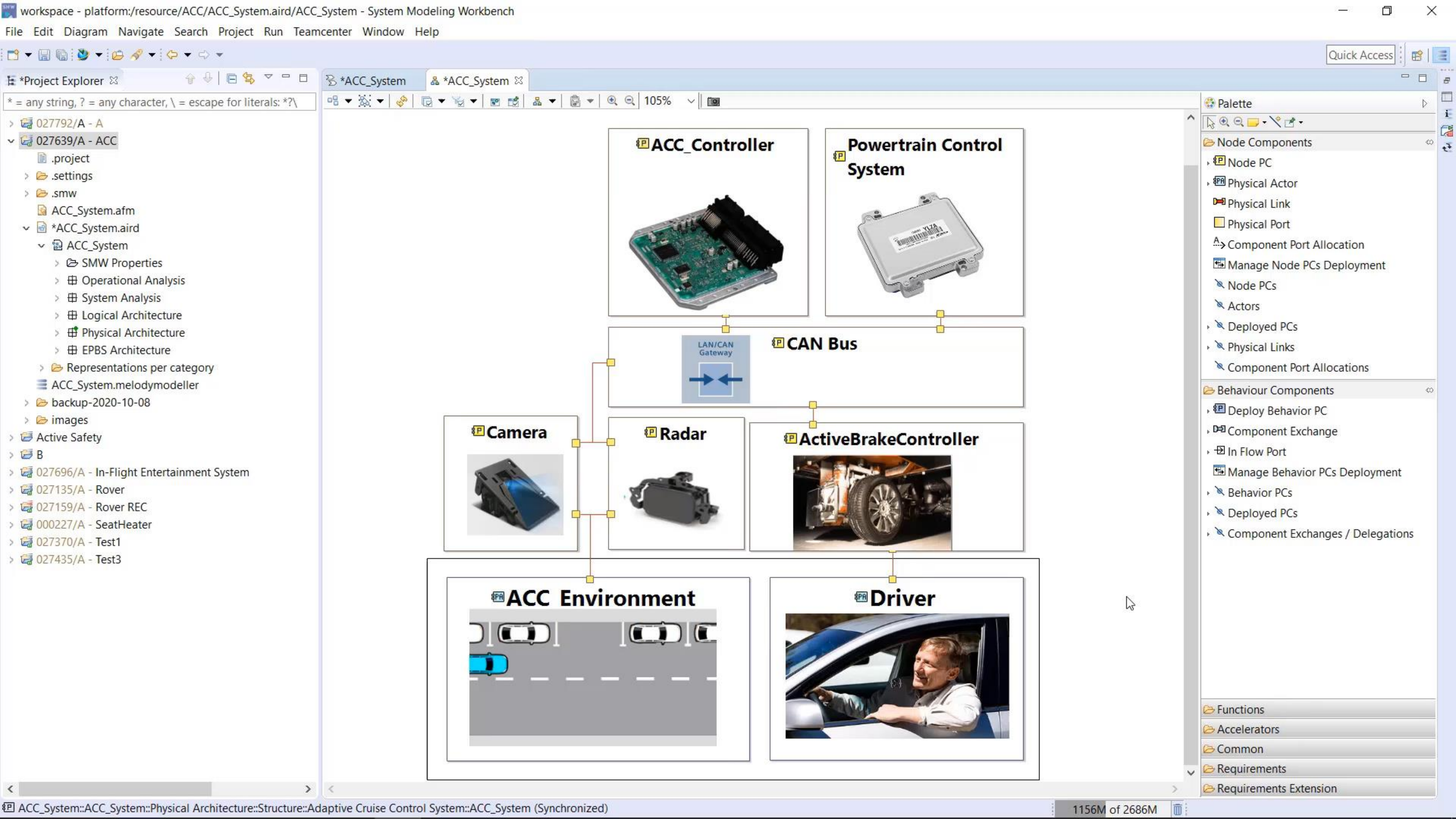
Requirements, Systems, Interfaces, Variability

Function Design, Traceability

Shared Product Architecture Variability

Teamcenter

Capital Systems Capture



workspace - platform:/resource/ACC/ACC_System.aird/ACC_System - System Modeling Workbench

ACC_System::ACC_System::Physical Architecture::Structure::Adaptive Cruise Control System::ACC_System (Synchronized)

Agenda

Methodology-Driven MBSE

Model functional-system architecture

Advanced systems management with product architecture

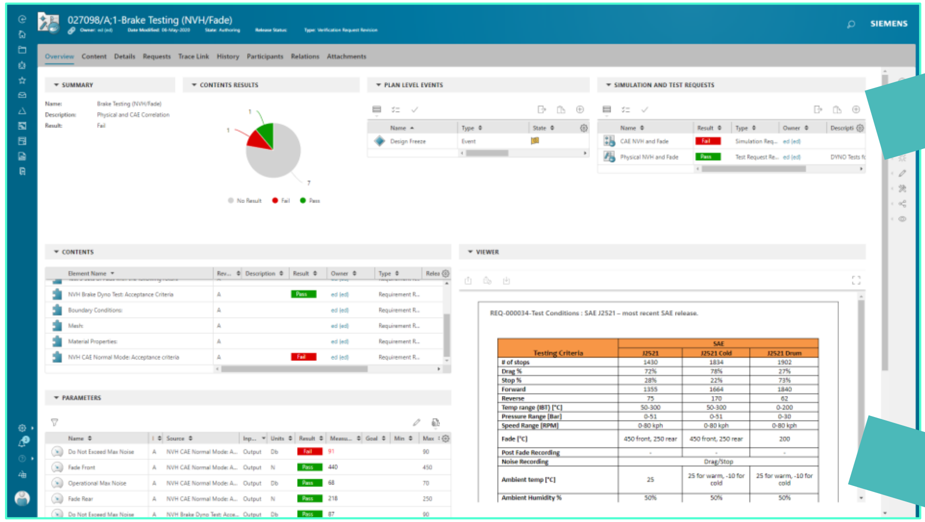
Re-use product architecture through multi-domain viewpoints

Close-the-loop with the product architecture

Verification Planning and Execution Management

Complete Verification Management for Simulation and Physical Testing

Verification Plan



027098/A;1-Brake Testing (NVH/Fade)

Overview | Content | Details | Requests | Trace Link | History | Participants | Relations | Attachments

SUMMARY

Name: Brake Testing (NVH/Fade)
Description: Physical and CAE Correlation
Result: Fail

CONTENTS RESULTS

PLAN LEVEL EVENTS

Name	Type	State
Design Freeze	Event	

SIMULATION AND TEST REQUESTS

Name	Result	Type	Owner	Descripti
CAE NVH and Fade	Fail	Simulation Req...	ed (ed)	
Physical NVH and Fade	Pass	Test Request Re...	ed (ed)	DYNO Tests fc

CONTENTS

Element Name	Rev...	Description	Result	Owner	Type
NVH Brake Dyno Test: Acceptance Criteria	A		Pass	ed (ed)	Requirement R...
Boundary Conditions:	A			ed (ed)	Requirement R...
Mesh:	A			ed (ed)	Requirement R...
Material Properties:	A			ed (ed)	Requirement R...
NVH CAE Normal Mode: Acceptance criteria	A		Fail	ed (ed)	Requirement R...

PARAMETERS

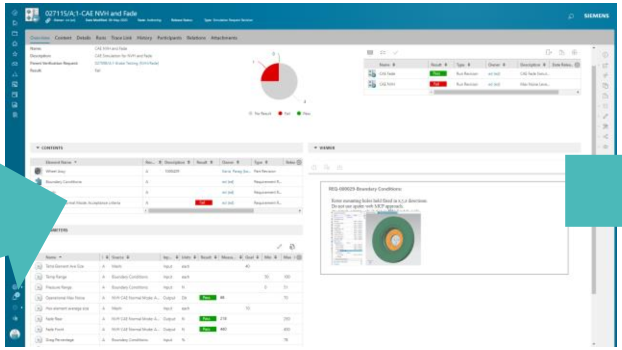
Name	I	Source	Inp...	Units	Result	Measu...	Goal	Min	Max
Do Not Exceed Max Noise	A	NVH CAE Normal Mode: A...	Output	Db	Fail	91			90
Fade Front	A	NVH CAE Normal Mode: A...	Output	N	Pass	440			450
Operational Max Noise	A	NVH CAE Normal Mode: A...	Output	Db	Pass	68			70
Fade Rear	A	NVH CAE Normal Mode: A...	Output	N	Pass	218			250
Do Not Exceed Max Noise	A	NVH Brake Dyno Test: Acce...	Output	Db	Pass	87			90

VIEWER

REQ-000034-Test Conditions : SAE J2521 – most recent SAE release.

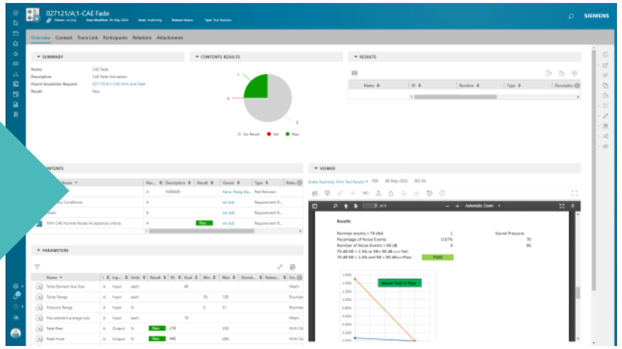
Testing Criteria	SAE		
	J2521	J2521 Cold	J2521 Drum
# of stops	1430	1834	1902
Drag %	72%	78%	27%
Stop %	28%	22%	73%
Forward	1355	1664	1840
Reverse	75	170	62
Temp range (IBT) [°C]	50-300	50-300	0-200
Pressure Range [Bar]	0-51	0-51	0-30
Speed Range [RPM]	0-80 kph	0-80 kph	0-80 kph
Fade [°C]	450 front, 250 rear	450 front, 250 rear	200
Post Fade Recording	-	-	-
Noise Recording	Drag/Stop		
Ambient temp [°C]	25	25 for warm, -10 for cold	25 for warm, -10 for cold
Ambient Humidity %	50%	50%	50%

Simulation Request



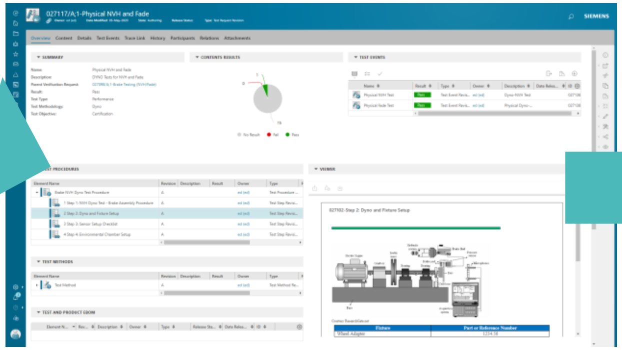
027115/A;1-CAE NVH and Fade

Simulation Run



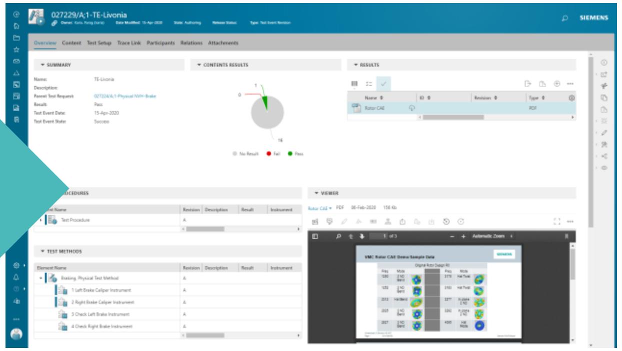
027121/A;1-CAE Fade

Test Request

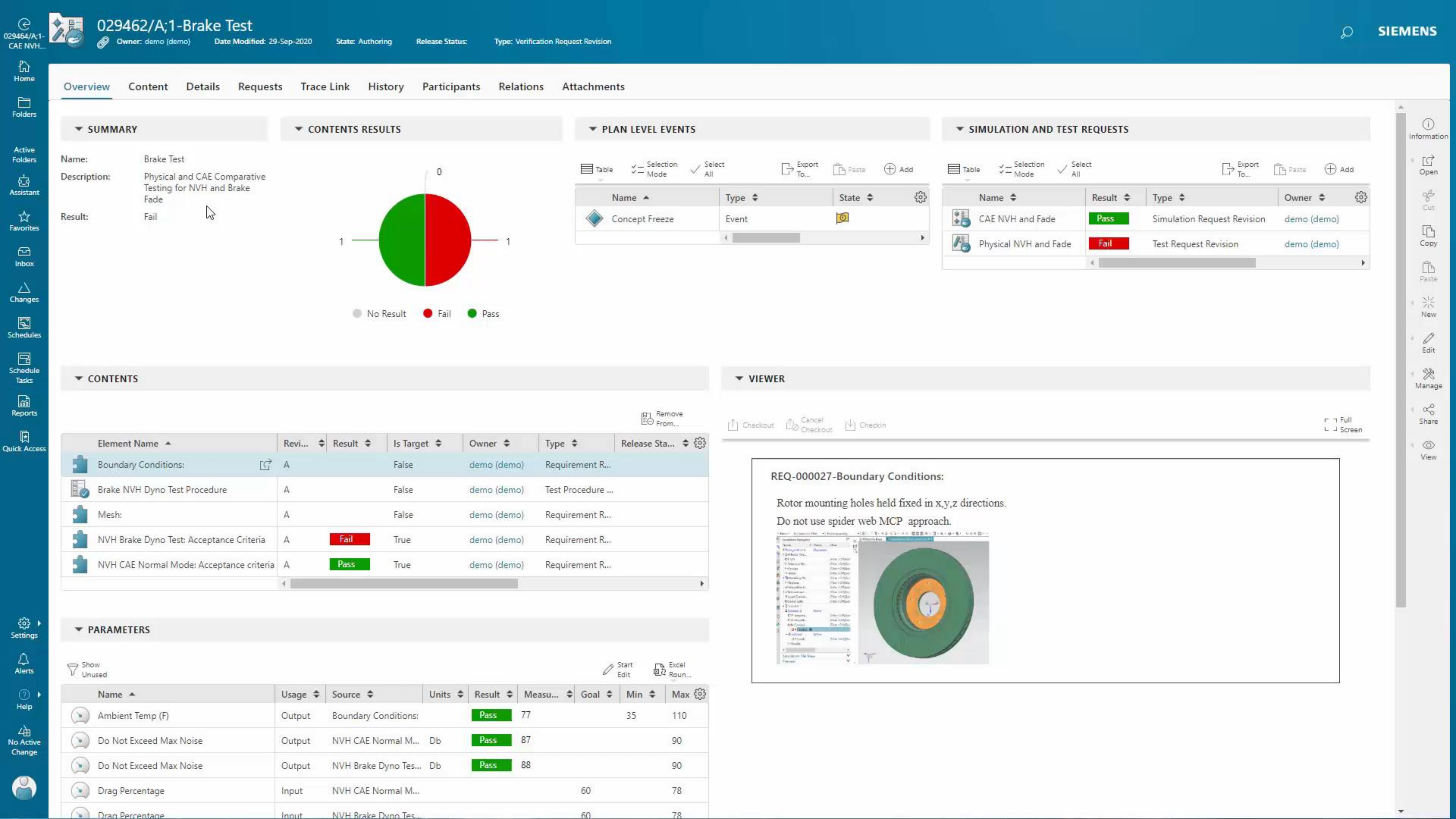


027117/A;1-Physical NVH and Fade

Test Run



027229/A;1-TE-Livonia



029462/A;1-Brake Test

Owner: demo (demo) Date Modified: 29-Sep-2020 State: Authoring Release Status: Type: Verification Request Revision

SUMMARY

Name: Brake Test

Description: Physical and CAE Comparative Testing for NVH and Brake Fade

Result: Fail

CONTENTS RESULTS

PLAN LEVEL EVENTS

Name	Type	State
Concept Freeze	Event	

SIMULATION AND TEST REQUESTS

Name	Result	Type	Owner
CAE NVH and Fade	Pass	Simulation Request Revision	demo (demo)
Physical NVH and Fade	Fail	Test Request Revision	demo (demo)

CONTENTS

Element Name	Revi...	Result	Is Target	Owner	Type	Release Sta...
Boundary Conditions:	A		False	demo (demo)	Requirement R...	
Brake NVH Dyno Test Procedure	A		False	demo (demo)	Test Procedure ...	
Mesh:	A		False	demo (demo)	Requirement R...	
NVH Brake Dyno Test: Acceptance Criteria	A	Fail	True	demo (demo)	Requirement R...	
NVH CAE Normal Mode: Acceptance criteria	A	Pass	True	demo (demo)	Requirement R...	

VIEWER

REQ-000027-Boundary Conditions:

Rotor mounting holes held fixed in x,y,z directions.

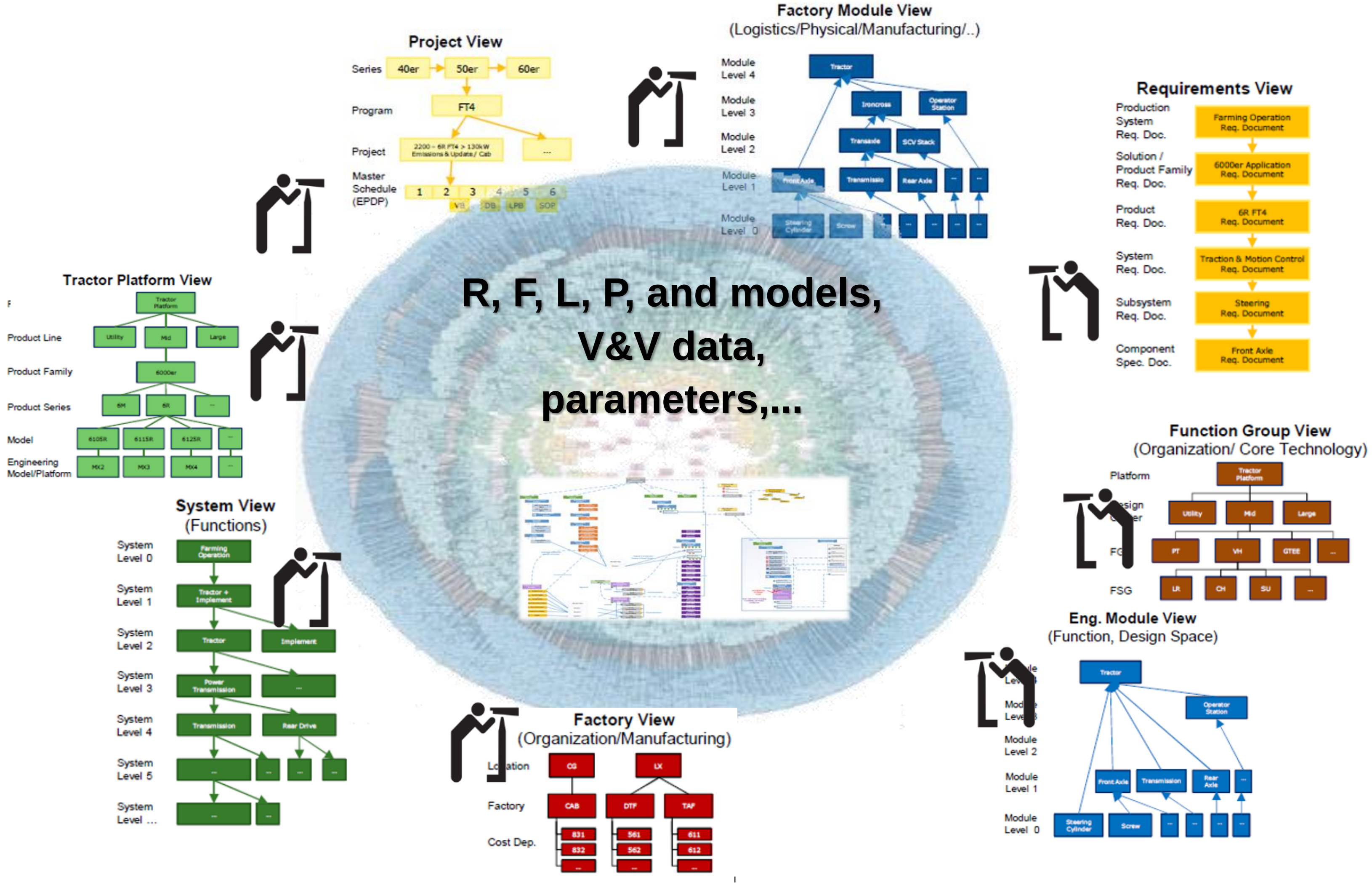
Do not use spider web MCP approach.

PARAMETERS

Name	Usage	Source	Units	Result	Measu...	Goal	Min	Max
Ambient Temp (F)	Output	Boundary Conditions:		Pass	77		35	110
Do Not Exceed Max Noise	Output	NVH CAE Normal M...	Db	Pass	87			90
Do Not Exceed Max Noise	Output	NVH Brake Dyno Tes...	Db	Pass	88			90
Drag Percentage	Input	NVH CAE Normal M...				60		78
Drag Percentage	Input	NVH Brake Dyno Tes...				60		78

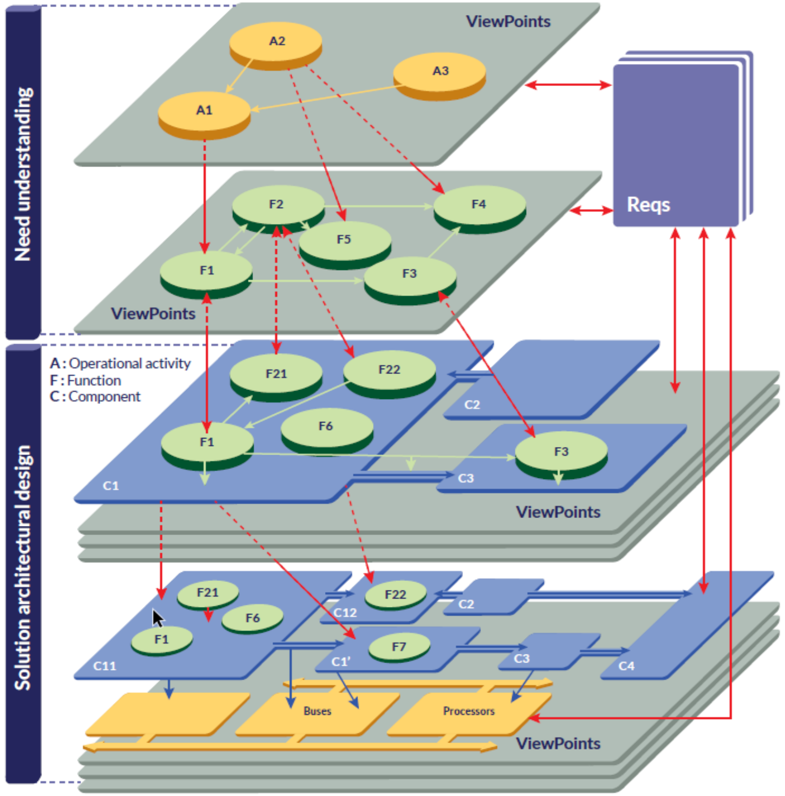
Eventually, everybody needs the system's information

- There are as many views as domains
- Multiple views within a same domain
- Views are "dynamic" to encompass continuous changes
- Views are automatically created based on criteria (configuration, scope, context, domain, …)



R, F, L, P, and models, V&V data, parameters,...

Open Source, Methodology-based, Adaptable, Supports SE Trajectory, Facilitates Collaboration, Works with Other Tools, Can start small ➔ Forward Compatible

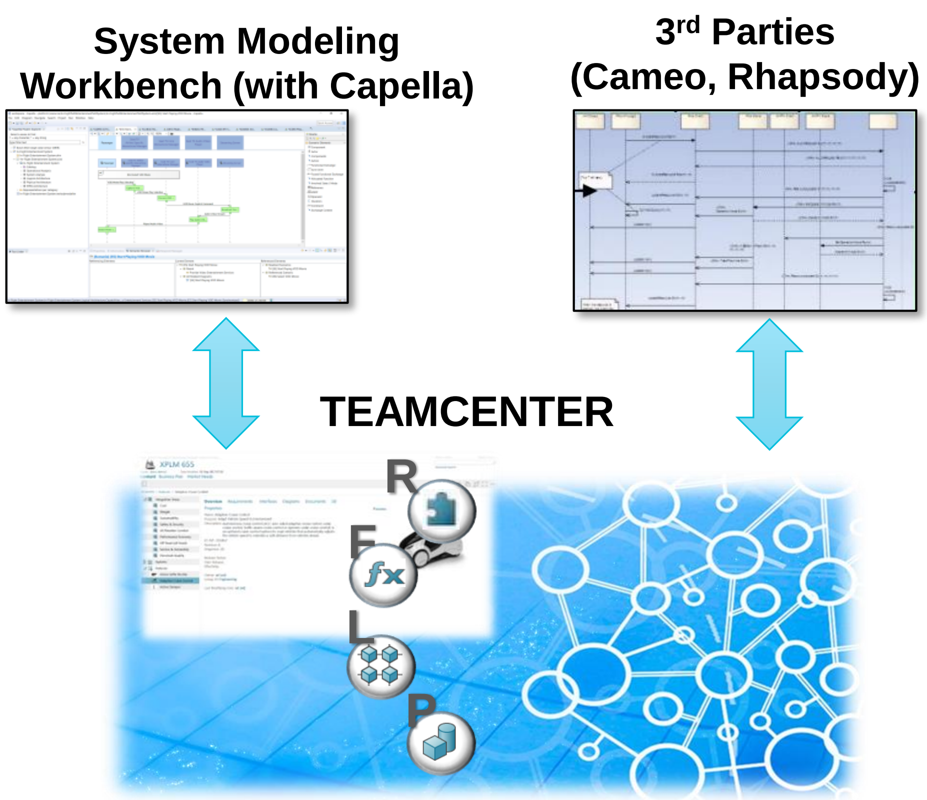


Operational Analysis
What the users of the system need to accomplish

Functional & Non Functional Need
What the system has to accomplish for the users

Logical Architecture
How the system will work to fulfill expectations

Physical Architecture
How the system will be developed and built


System Modeling Workbench (with Capella)
3rd Parties (Cameo, Rhapsody)
TEAMCENTER
R
F
L
P

Try it yourself. Capella version 5.1 deploys completely without added set-up on Windows, Mac and Linux.
https://www.eclipse.org/capella/download.html

Learn more about forward compatibility at https://www.plm.automation.siemens.com/global/en/webinar/integrated-mbse/85390 (or contact me)

Contact

Lisa Murphy
Digital Transformer, Aerospace Defense Federal & Marine
Siemens Digital Industries Software
Atlanta, GA USA

Mobile +1 770 548 5225

E-mail: lisa.murphy@siemens.com

Disclaimer

Subject to changes and errors. The information given in this document only contains general descriptions and/or performance features which may not always specifically reflect those described, or which may undergo modification in the course of further development of the products. The requested performance features are binding only when they are expressly agreed upon in the concluded contract.

All product designations may be trademarks or other rights of Siemens AG, its affiliated companies or other companies whose use by third parties for their own purposes could violate the rights of the respective owner.