32nd Annual INCOSE
international symposium
hybrid event
Detroit, MI, USA
June 25 - 30, 2022

A Platform for MBSE-Enabled, Digitally Threaded, Electronics Design and Verification

SE Vision: Connected Engineering of Systems Across the Lifecycle

REQUIRED BEHAVIORAL STRUCTURAL AS DESIGNED AS ORDERED AS BUILT AS DELIVERED AS SERVICED

Systems Engineering

- *Begins at the conceptual design phase, continuing throughout the life cycle*
- *Defines and validates requirements to meet user needs*
- *Designs, analyzes and verifies a system to meet the requirements*

Before there was CAD/CAE, there was Systems Engineering. Many types of SE documents and diagrams were produced, maintained and shared manually.

If MBSE is to be CAD/CAE for SE, are we there yet?

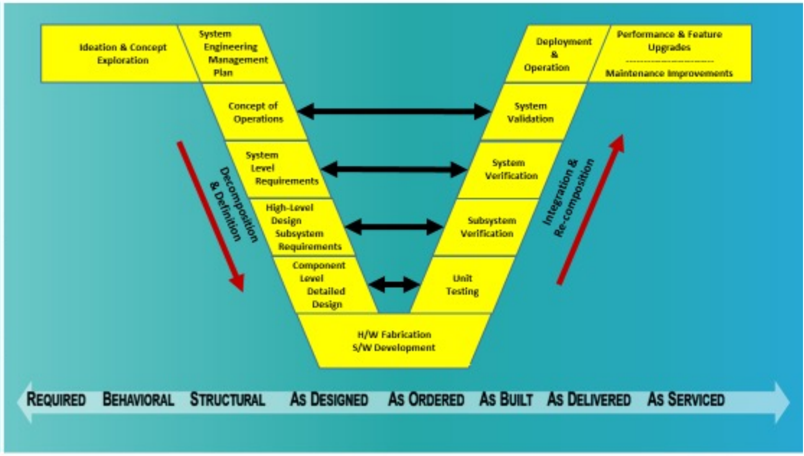
Engineering "V" – Emphasis on Continuous Verification Levels

Verification requires feedback loops in the life cycle time line

The engineering "V" diagram emphasizes verification feedback and detail layers

Requirement verification plans MUST be continuously refined and then performed EARLY

SE work-products must share architecture info to express requirements to be verified


Ideation & Concept Exploration
System Engineering Management Plan
Concept of Operations
System Level Requirements
High-Level Design Subsystem Requirements
Component Level Detailed Design
H/W Fabrication S/W Development
Unit Testing
Subsystem Verification
System Verification
System Validation
Deployment & Operation
Performance & Feature Upgrades
Maintenance Improvements
Decomposition & Definition
Integration & Re-composition
REQUIRED
BEHAVIORAL
STRUCTURAL
AS DESIGNED
AS ORDERED
AS BUILT
AS DELIVERED
AS SERVICED

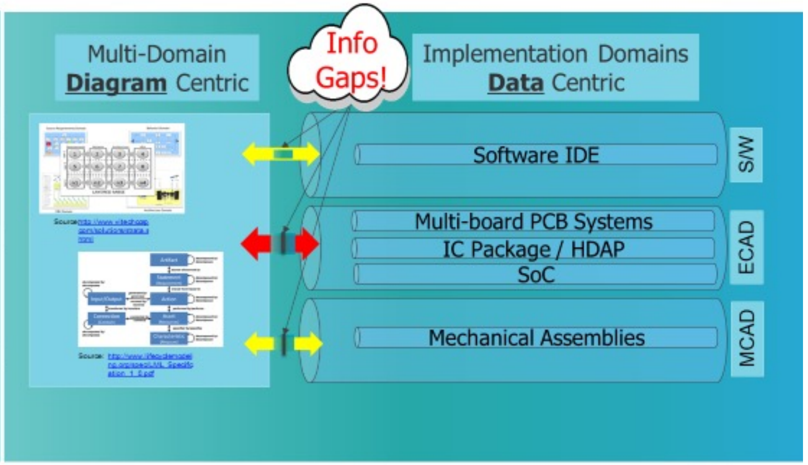
Current Electronics Status: MBSE-driven Architecture is NOT SHAREABLE;

Most SE diagrams are not intuitive. Domain engineers are not fluent in them.

Most diagrams are not stored as data elements in enterprise level database repositories.

Gaps exist which restrict the flow of information. The ECAD gap is the most severe.

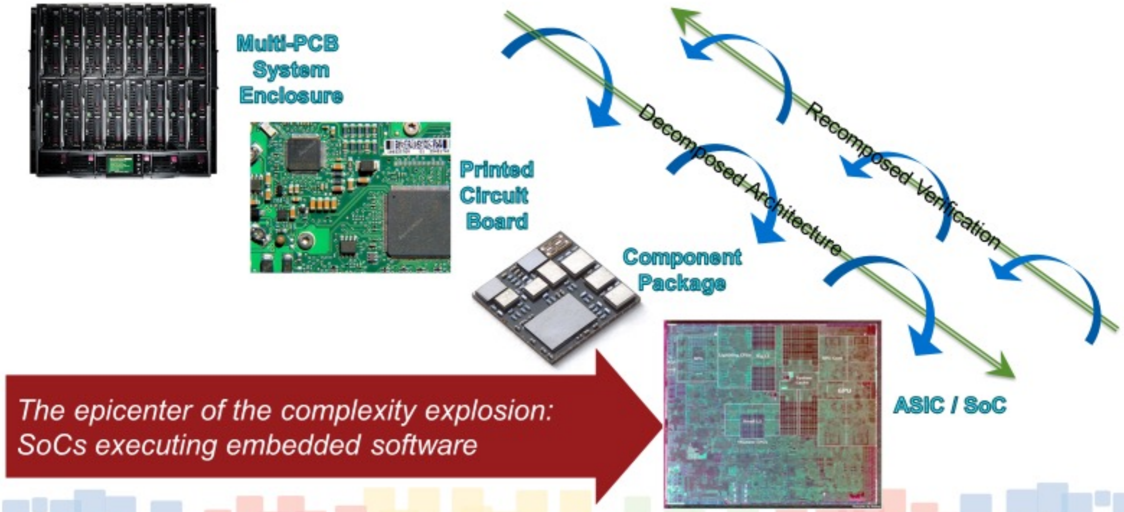
ECAD has sub-domain decomposition layers and the deepest verification challenge.



Source: http://www.vitechcorp.com/solutions/[illegible].html

Source: http://www.lifecyclemodeling.org/spec/LML_Specification_1_0.pdf

Electronics Requires Continuous Decomposition and Verification to Realize "System-to-Silicon" Verification

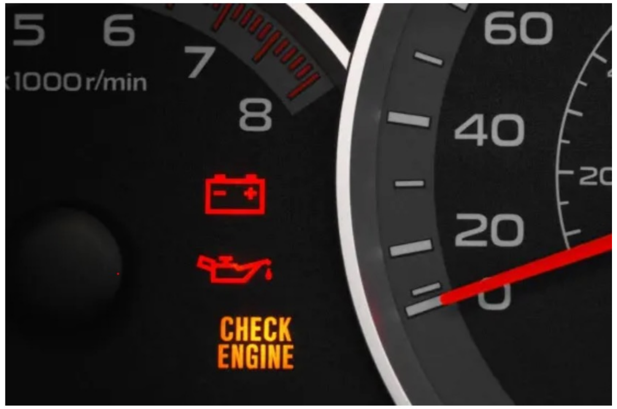


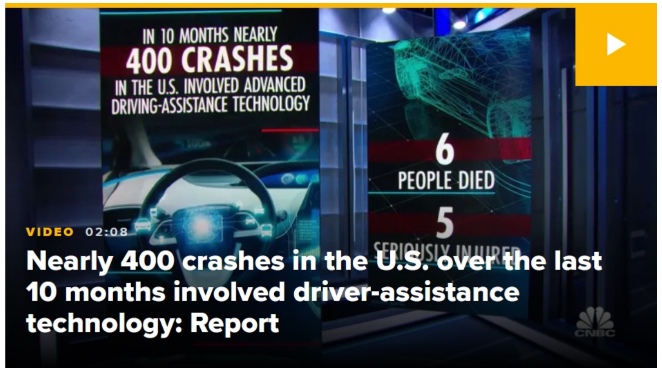
The epicenter of the complexity explosion: SoCs executing embedded software

Risks of Functional Allocation Overloading Electronics

Then

Now





IN 10 MONTHS NEARLY 400 CRASHES IN THE U.S. INVOLVED ADVANCED DRIVING-ASSISTANCE TECHNOLOGY

6 PEOPLE DIED

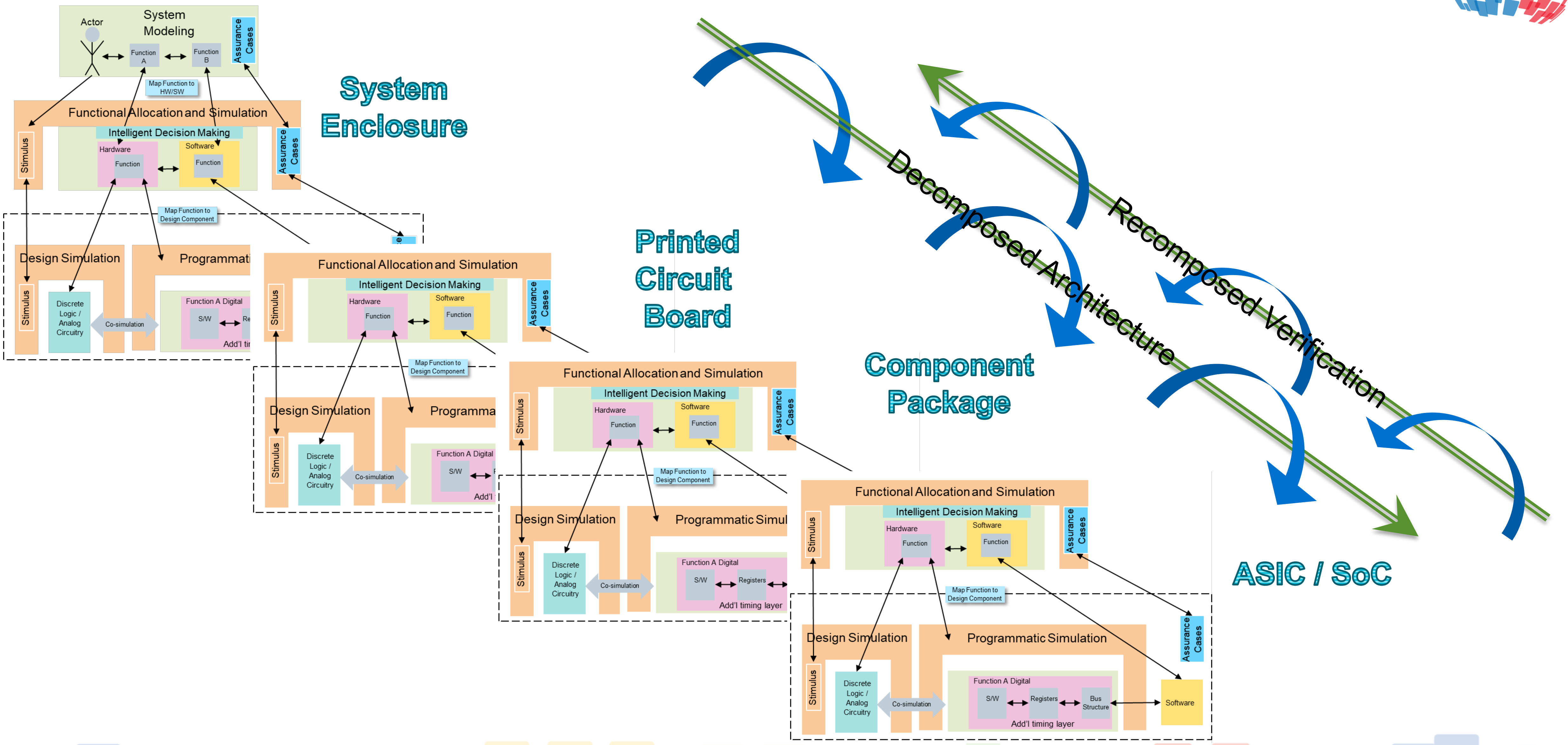
5 SERIOUSLY INJURED

VIDEO 02:08

Nearly 400 crashes in the U.S. over the last 10 months involved driver-assistance technology: Report

Soon?

Future State: Commonality Across Workflows, System-to-Silicon



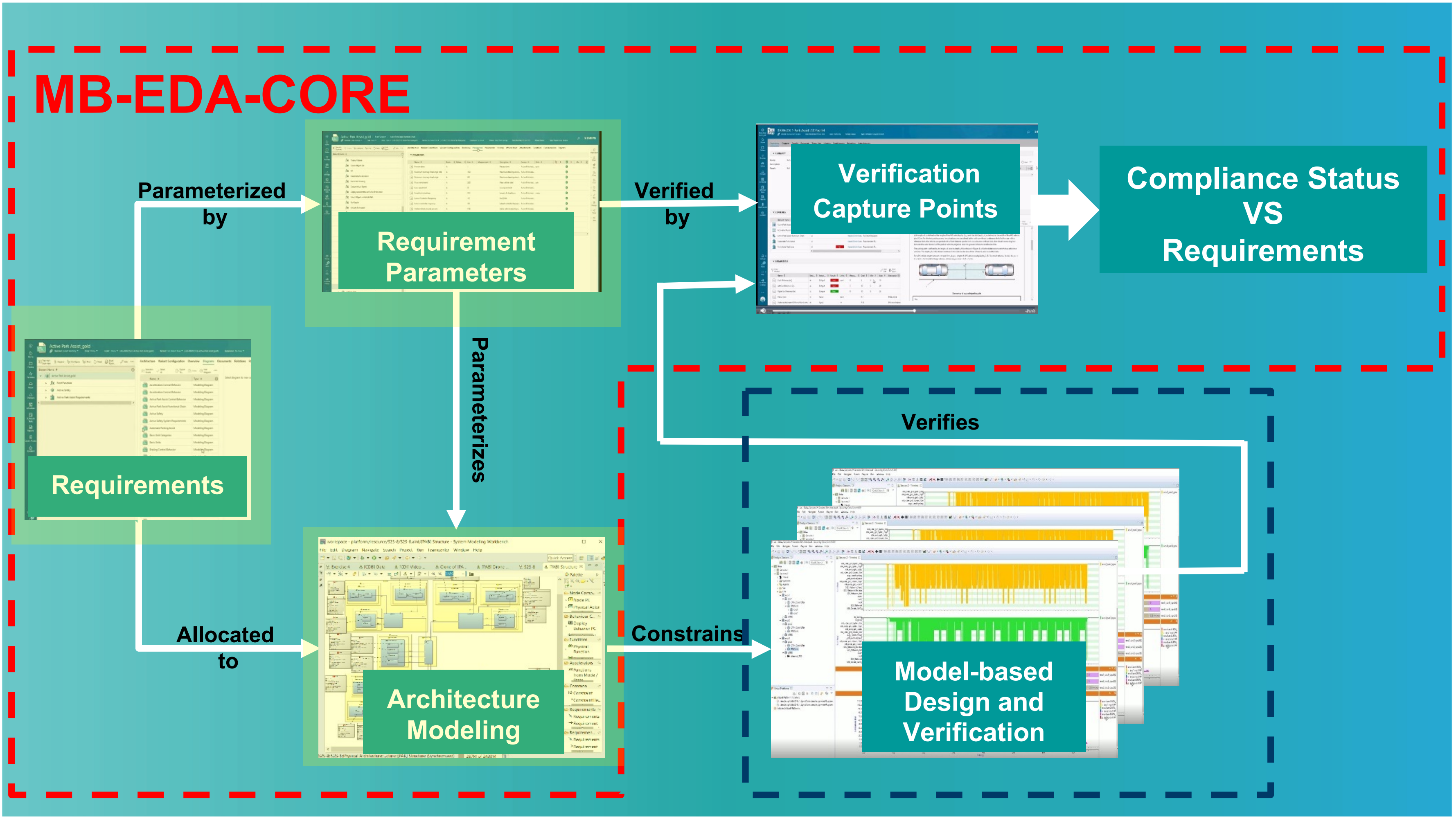
A Solution Pattern Implementation Consists of both Domain Independent (ASoT) and Domain/Sub-domain Dependent Workflow Components and Tools

Enabling the ASoT can be domain independent and non-ECAD specific

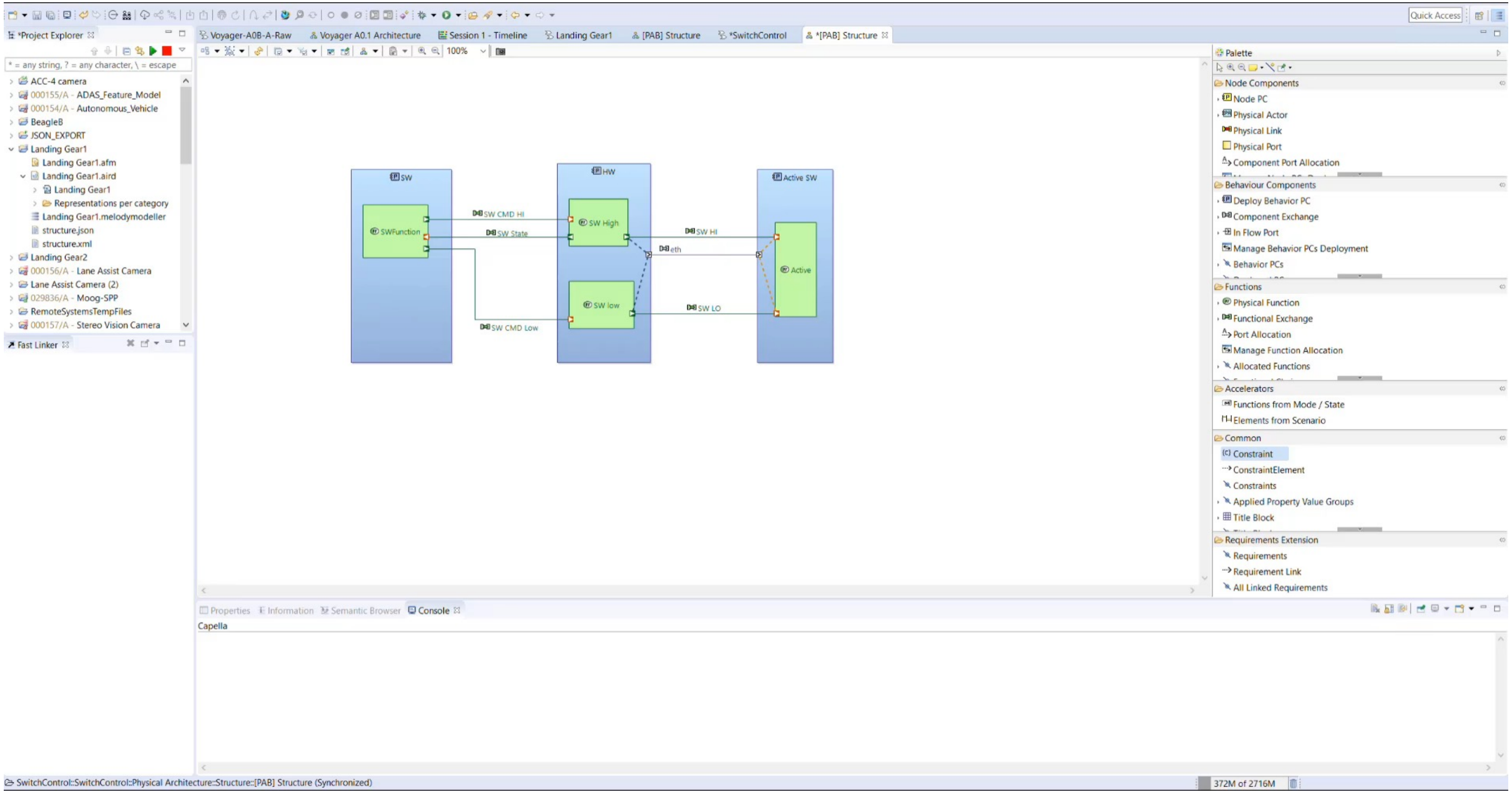
Domain specific tools can potentially be any design and verification toolset

Domain specific tools will likely need automation to increase efficiency / reduce cycle time

Bottom-up approach: Implement VCPs for what is simulated today



Requirement Allocation Example



Capella

SwitchControl::SwitchControl::Physical Architecture::Structure::[PAB] Structure (Synchronized)

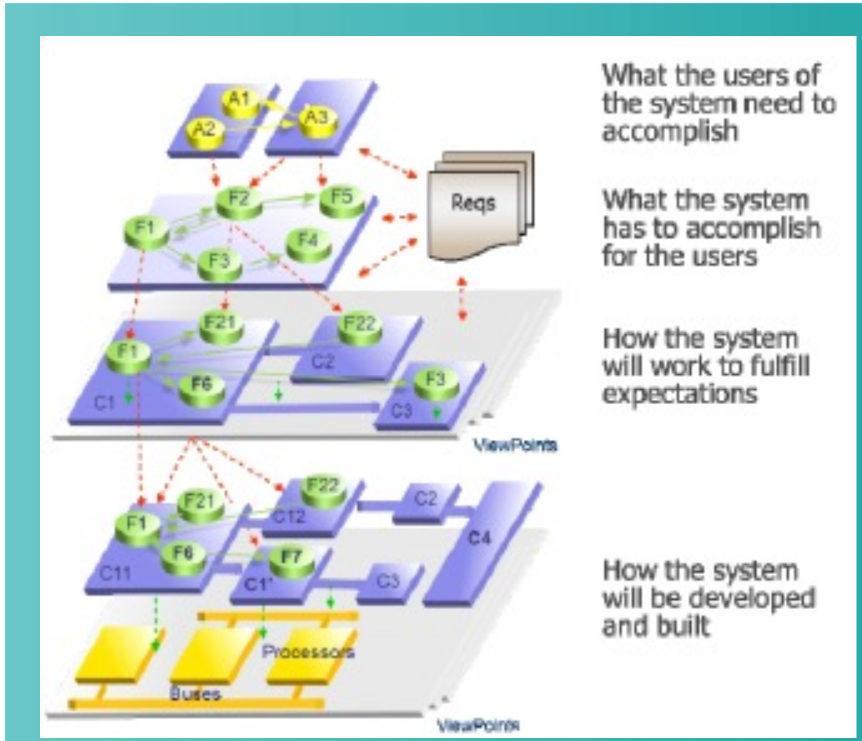
ARCADIA/Capella Uniquely Addresses Continuous Decomposition

Siemens selected Arcadia/Capella as basis for System Modeling Workbench

Most modern method and tooling with most advanced refinement capabilities

CAD/CAE style of underlying data model

Guides the user to assure models are correct and well formed

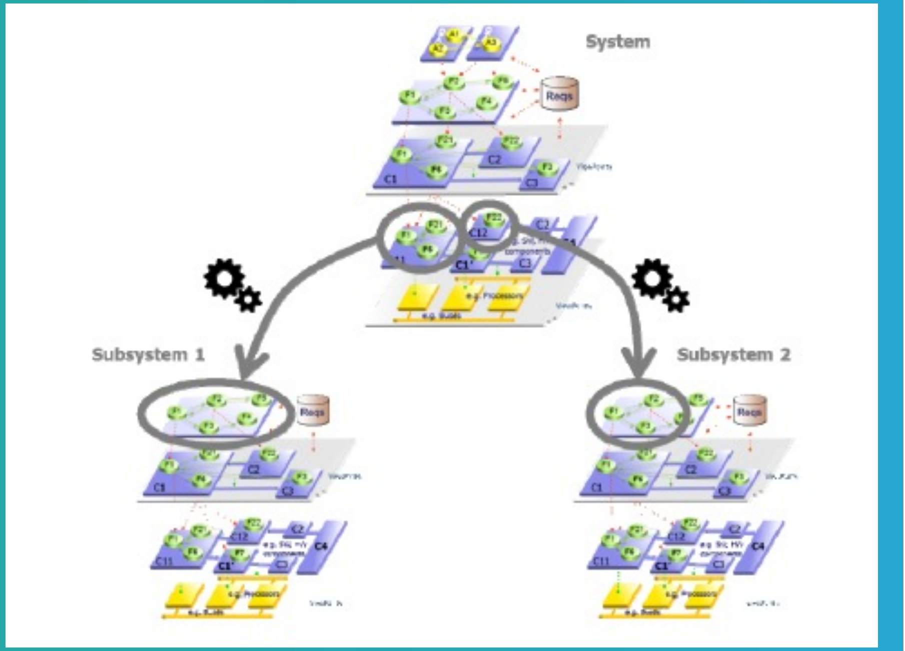


What the users of the system need to accomplish

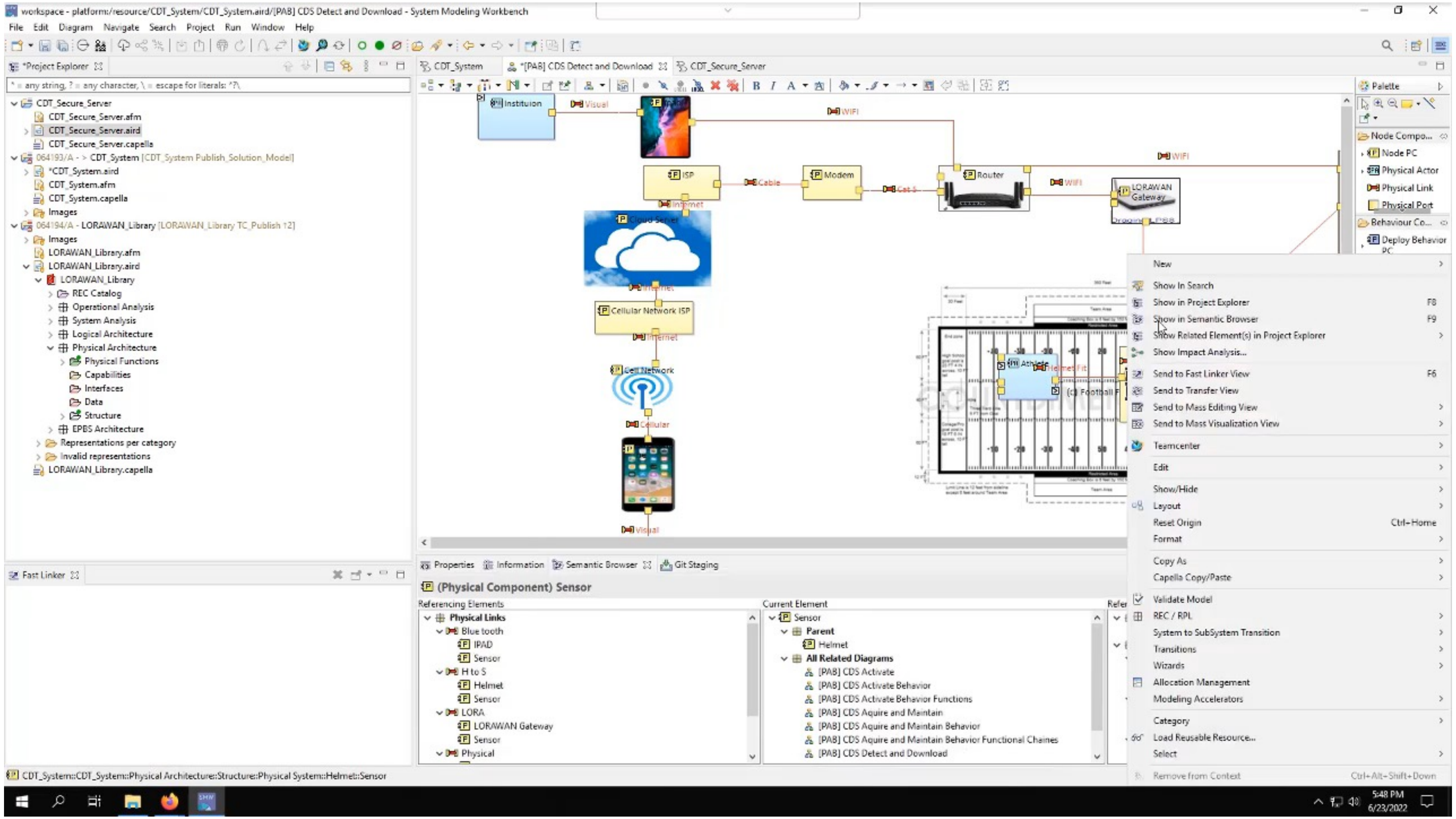
What the system has to accomplish for the users

How the system will work to fulfill expectations

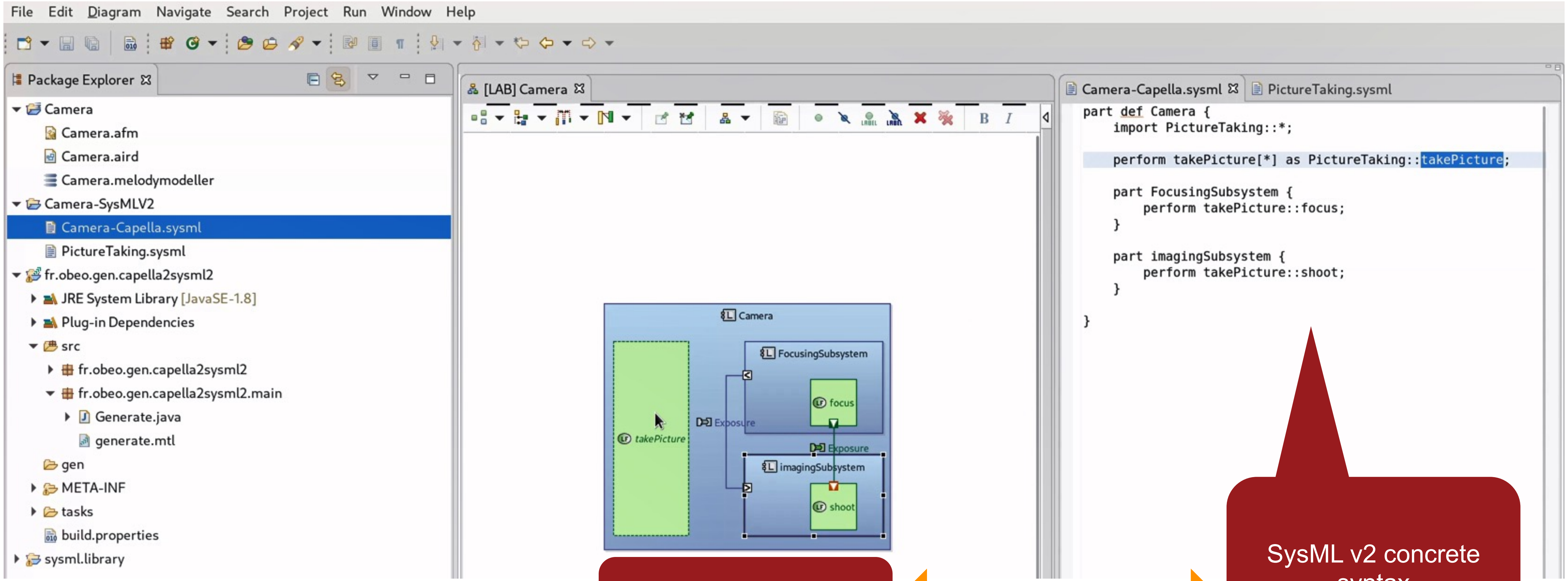
How the system will be developed and built



Sub-system transition



Path to Standards Conformance: Working toward Capella Generative SysML V2 Concrete Syntax



- Camera
  - Camera.afm
  - Camera.aird
  - Camera.melodymodeller
- Camera-SysMLV2
  - Camera-Capella.sysml
  - PictureTaking.sysml
- fr.obeo.gen.capella2sysml2
  - JRE System Library [JavaSE-1.8]
  - Plug-in Dependencies
  - src
    - fr.obeo.gen.capella2sysml2
    - fr.obeo.gen.capella2sysml2.main
      - Generate.java
      - generate.mtl
  - gen
  - META-INF
  - tasks
  - build.properties
- sysml.library

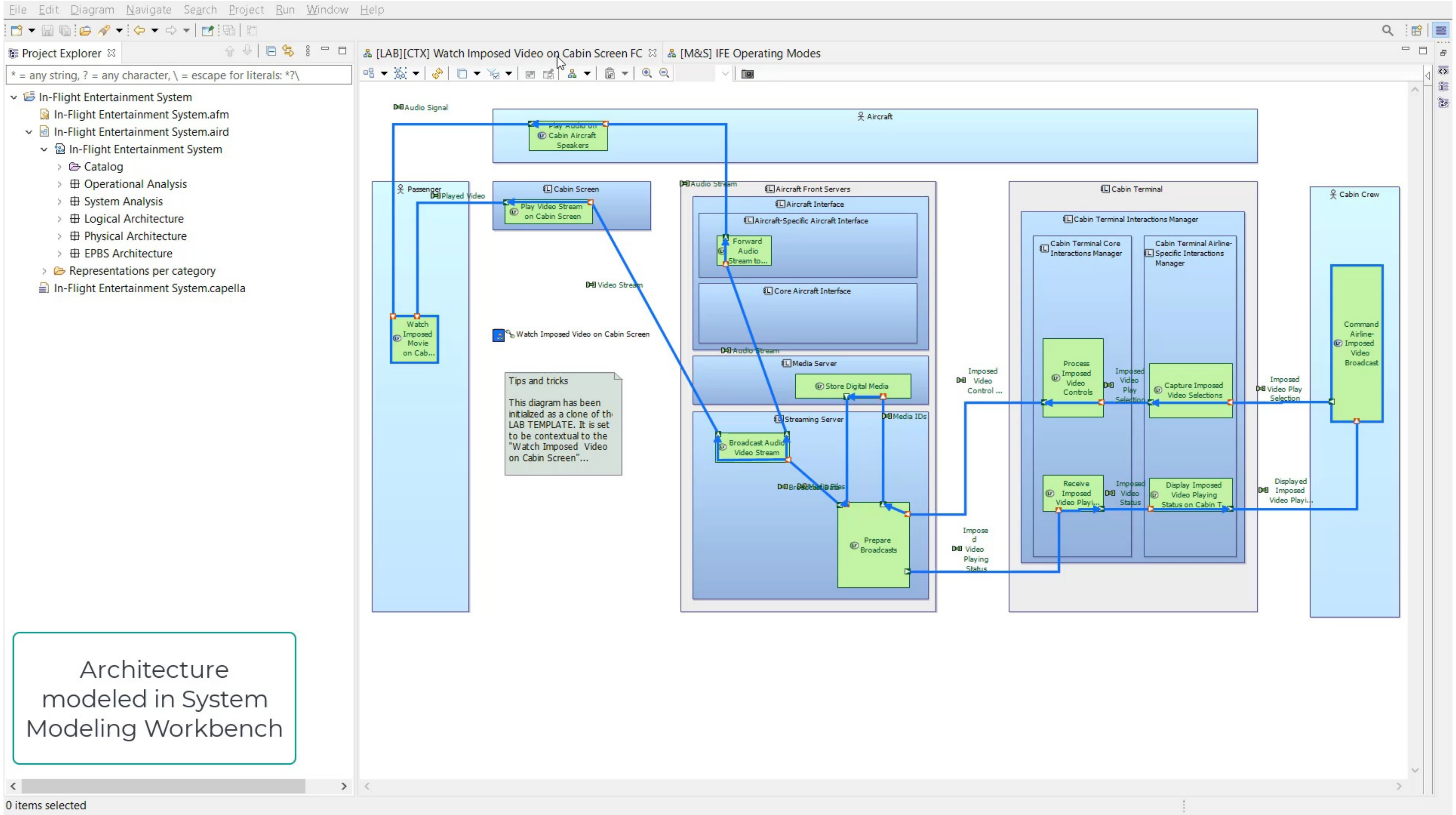
```
part def Camera {
    import PictureTaking::*;

    perform takePicture[*] as PictureTaking::takePicture;

    part FocusingSubsystem {
        perform takePicture::focus;
    }

    part imagingSubsystem {
        perform takePicture::shoot;
    }

}
```

SysML v2 concrete syntax

Logical Architecture

Generative

SysML V2 Example


File
Edit
Diagram
Navigate
Search
Project
Run
Window
Help
Project Explorer
* = any string, ? = any character, \ = escape for literals: *?\
In-Flight Entertainment System
In-Flight Entertainment System.afm
In-Flight Entertainment System.aird
In-Flight Entertainment System
Catalog
Operational Analysis
System Analysis
Logical Architecture
Physical Architecture
EPBS Architecture
Representations per category
In-Flight Entertainment System.capella
[LAB][CTX] Watch Imposed Video on Cabin Screen FC
[M&S] IFE Operating Modes
Audio Signal
Aircraft
Play Audio on Cabin Aircraft Speakers
Passenger
Played Video
Cabin Screen
Play Video Stream on Cabin Screen
Audio Stream
Aircraft Front Servers
Aircraft Interface
Aircraft-Specific Aircraft Interface
Forward Audio Stream to...
Core Aircraft Interface
Video Stream
Watch Imposed Movie on Cab...
Watch Imposed Video on Cabin Screen
Audio Stream
Media Server
Store Digital Media
Streaming Server
Media IDs
Broadcast Audio Video Stream
Broadcast Data
Media Files
Prepare Broadcasts
Imposed Video Control ...
Imposed Video Playing Status
Cabin Terminal
Cabin Terminal Interactions Manager
Cabin Terminal Core Interactions Manager
Cabin Terminal Airline-Specific Interactions Manager
Process Imposed Video Controls
Imposed Video Play Selection
Capture Imposed Video Selections
Receive Imposed Video Playi...
Imposed Video Status
Display Imposed Video Playing Status on Cabin T...
Cabin Crew
Command Airline-Imposed Video Broadcast
Imposed Video Play Selection
Displayed Imposed Video Playi...
0 items selected

Tips and tricks

This diagram has been initialized as a clone of th LAB TEMPLATE. It is set to be contextual to the "Watch Imposed Video on Cabin Screen"...

Architecture modeled in System Modeling Workbench

Allocatable Electronics Performance Levels

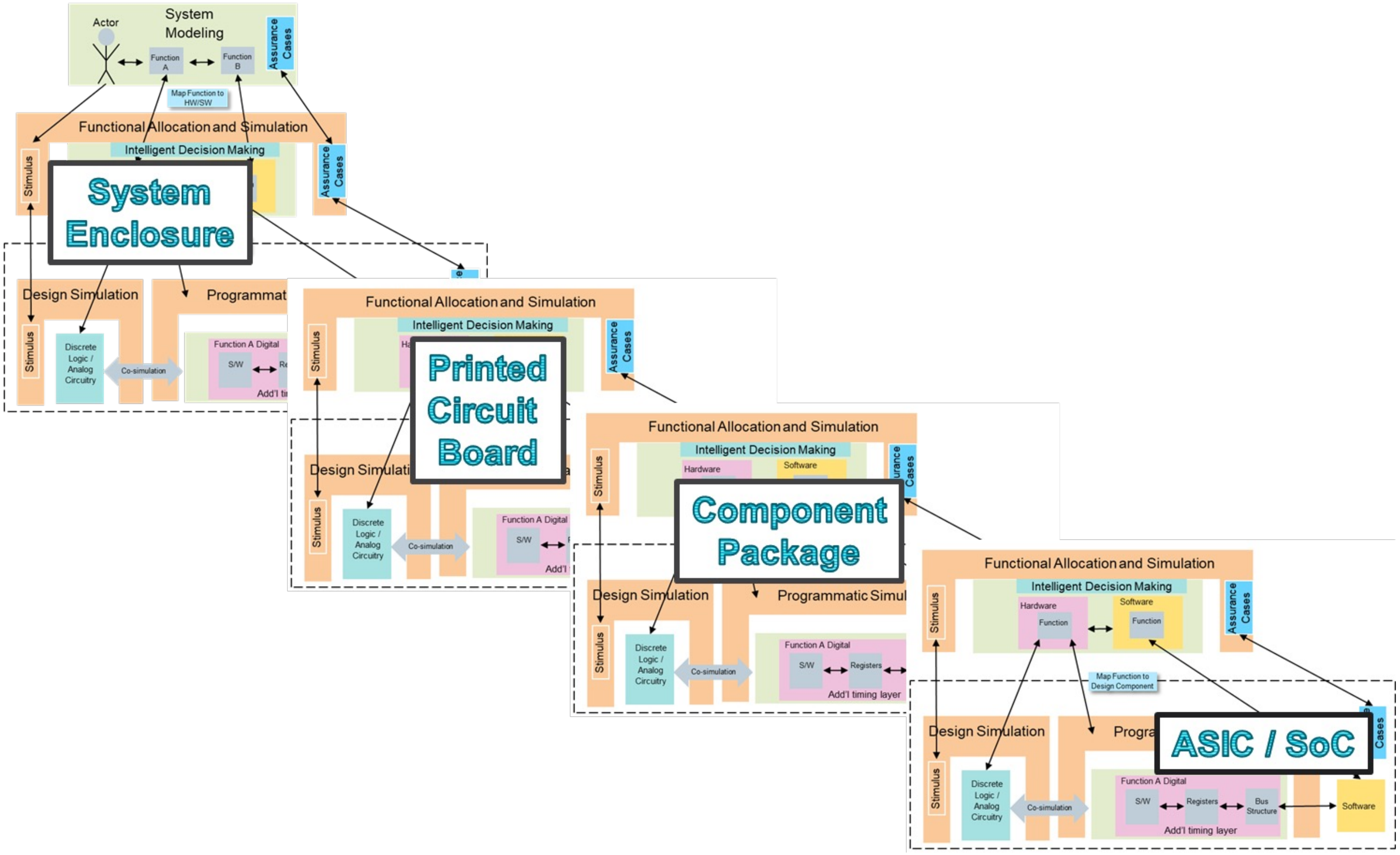
1. SoC INDEPENDENT s/w

2. SoC Dependent S/W
(e.g. CUDA, Assembler)

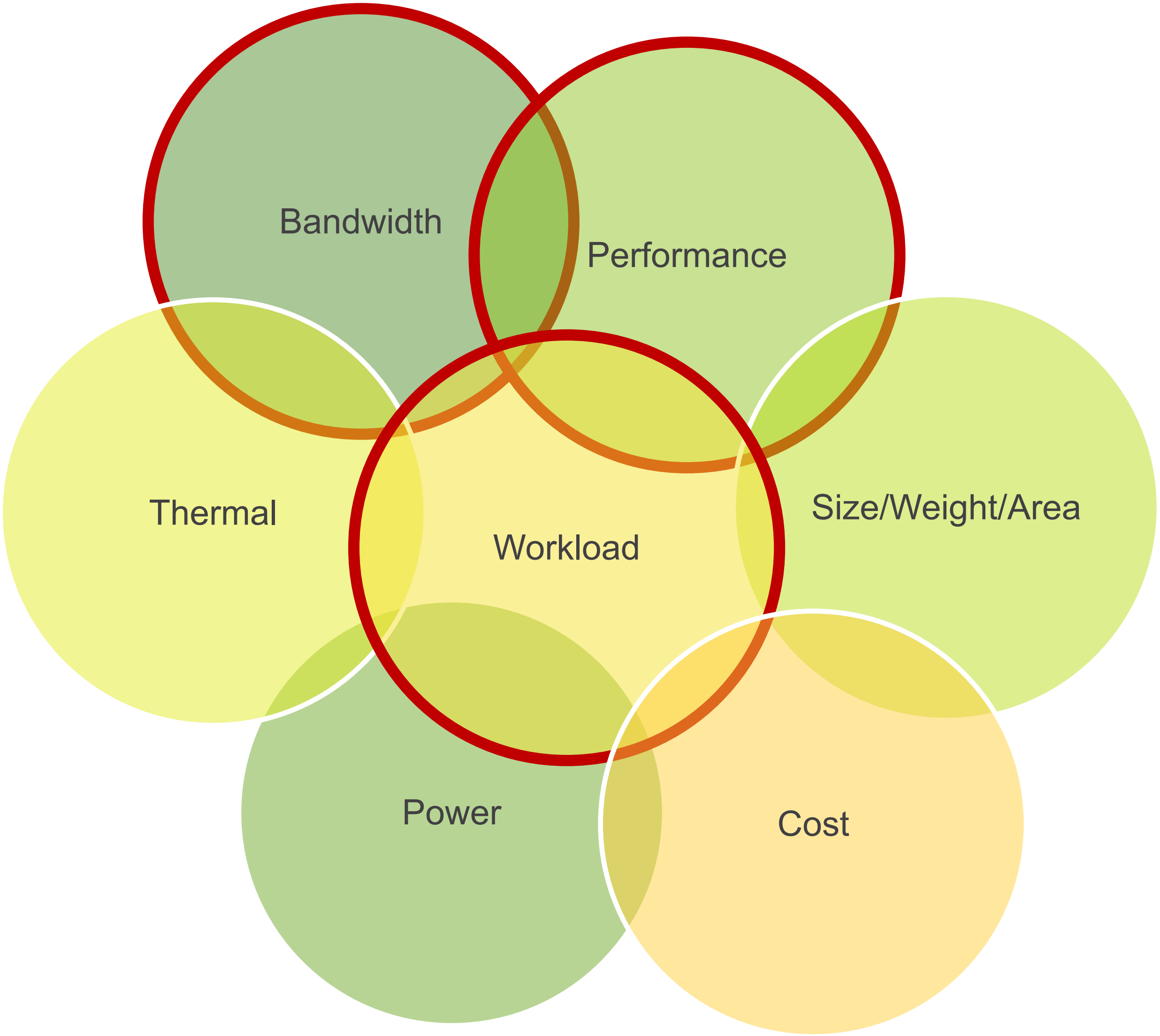
3. Embedded Processor IP Dependent S/W

4. H/W Accelerator Plus Embedded Processor IP and Dependent S/W

5. Native H/W Acceleration

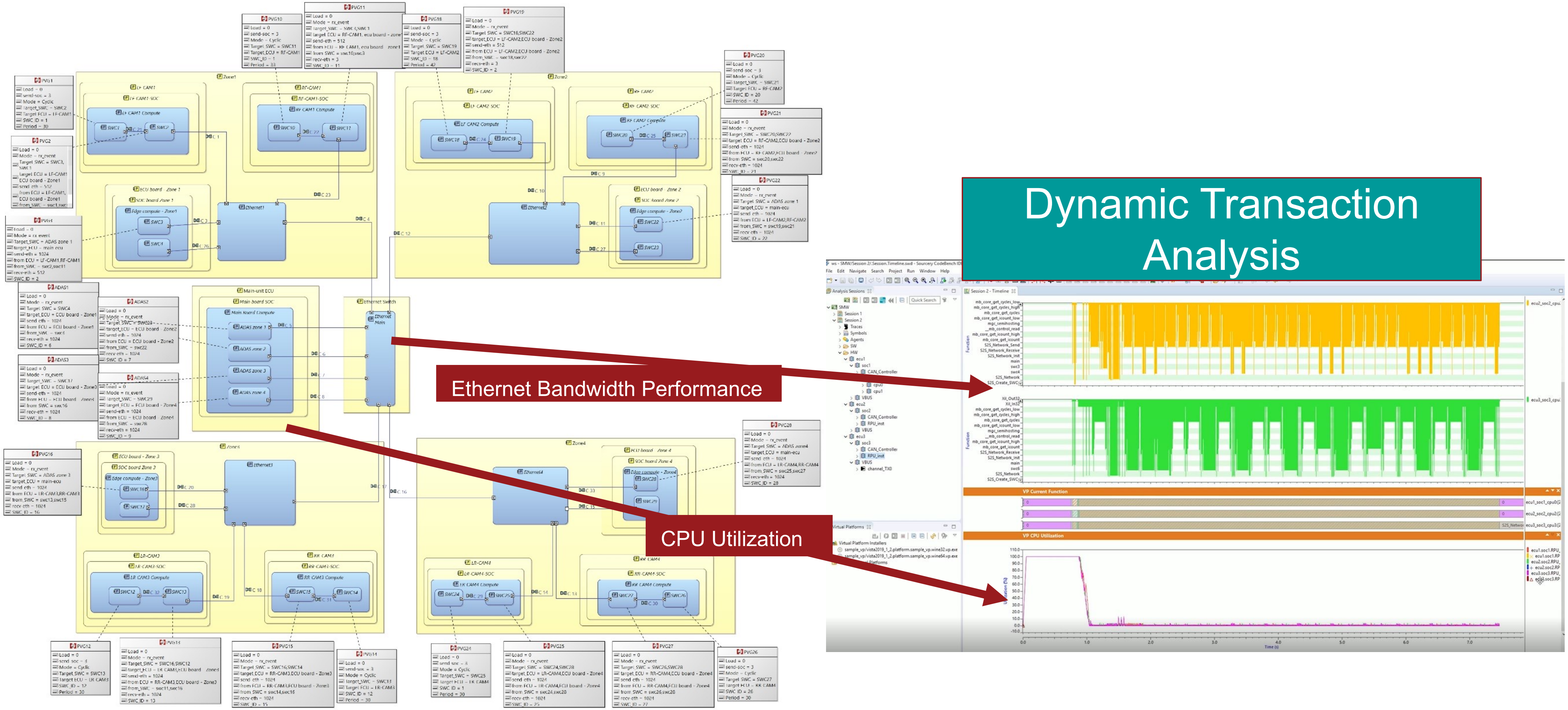


Architecture Design Options are Expanding Exponentially



Approximate the electronics to narrow the tradeoff space

Architecture Exploration by Transaction Level Simulation; System-to-Silicon

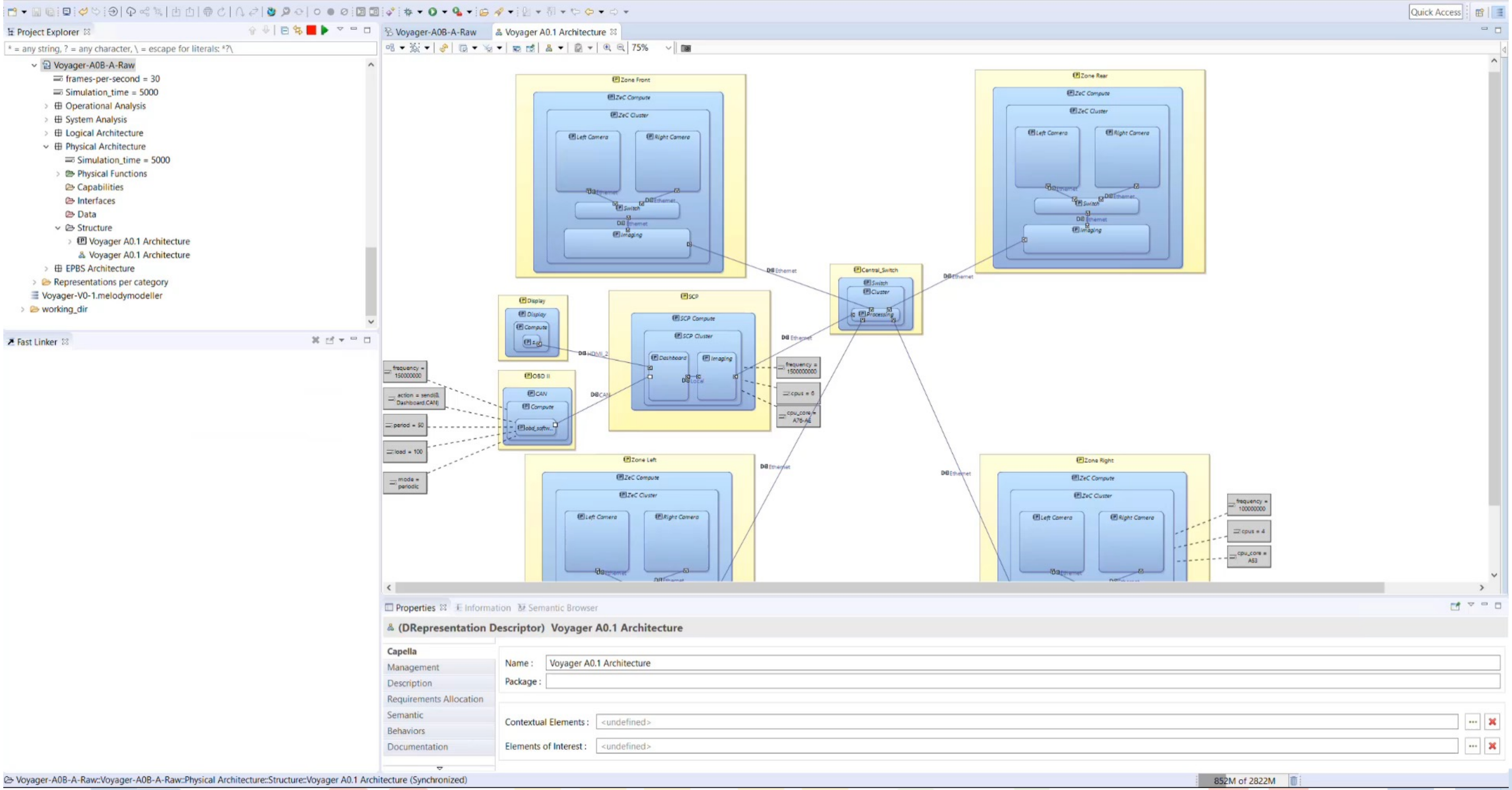


Dynamic Transaction Analysis

Ethernet Bandwidth Performance

CPU Utilization

Architecture Analysis Example



Project Explorer

* = any string, ? = any character, \ = escape for literals: *?\

- Voyager-A0B-A-Raw
  - frames-per-second = 30
  - Simulation_time = 5000
  - Operational Analysis
  - System Analysis
  - Logical Architecture
  - Physical Architecture
    - Simulation_time = 5000
    - Physical Functions
    - Capabilities
    - Interfaces
    - Data
    - Structure
      - Voyager A0.1 Architecture
      - Voyager A0.1 Architecture
  - EPBS Architecture
  - Representations per category
- Voyager-V0-1.melodymodeller
- working_dir

Fast Linker

Voyager-A0B-A-Raw | Voyager A0.1 Architecture

75%

Properties | Information | Semantic Browser

(DRepresentation Descriptor) Voyager A0.1 Architecture

Capella

Management

Description

Requirements Allocation

Semantic

Behaviors

Documentation

Name : Voyager A0.1 Architecture

Package :

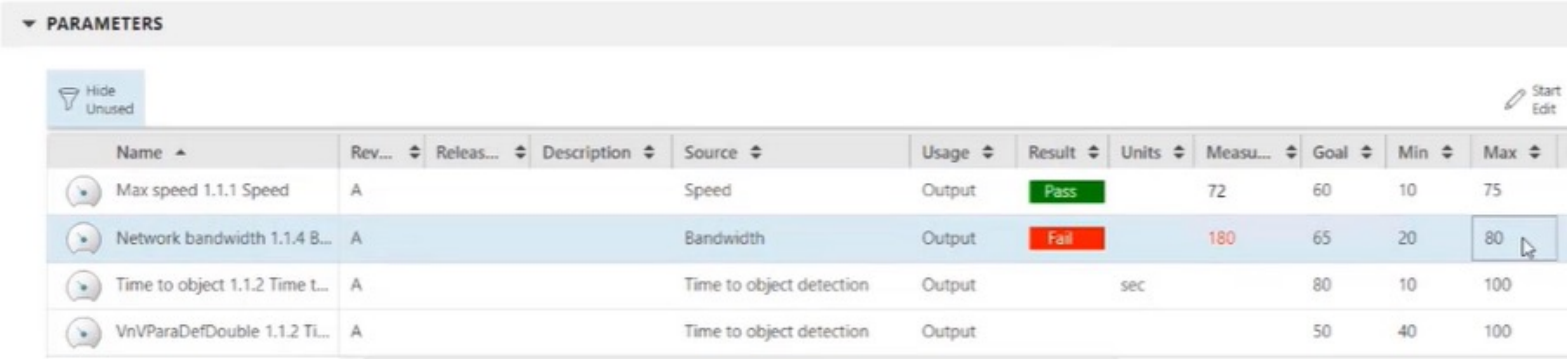
Contextual Elements : <undefined>

Elements of Interest : <undefined>

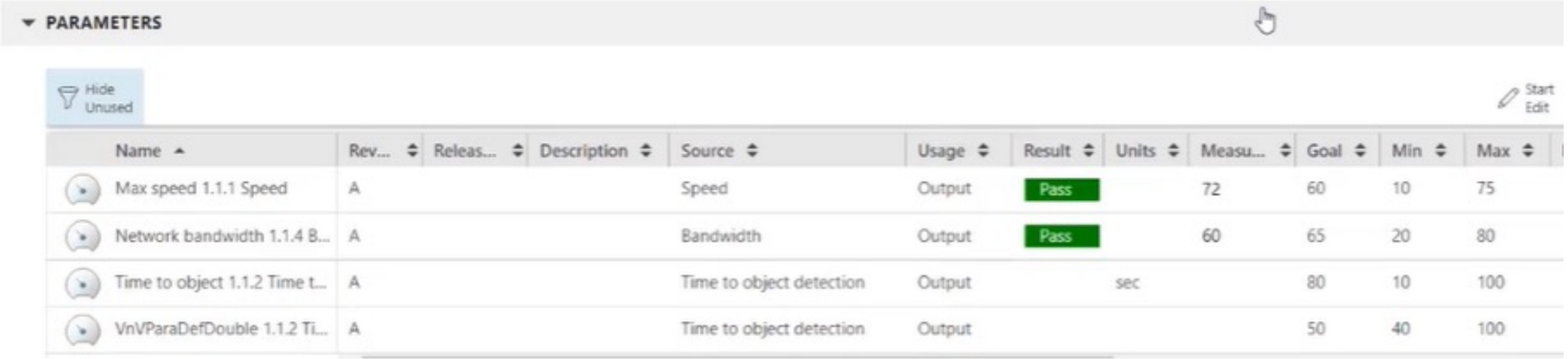
Voyager-A0B-A-Raw::Voyager-A0B-A-Raw::Physical Architecture::Structure::Voyager A0.1 Architecture (Synchronized)

852M of 2822M

Verification Capture Point Example



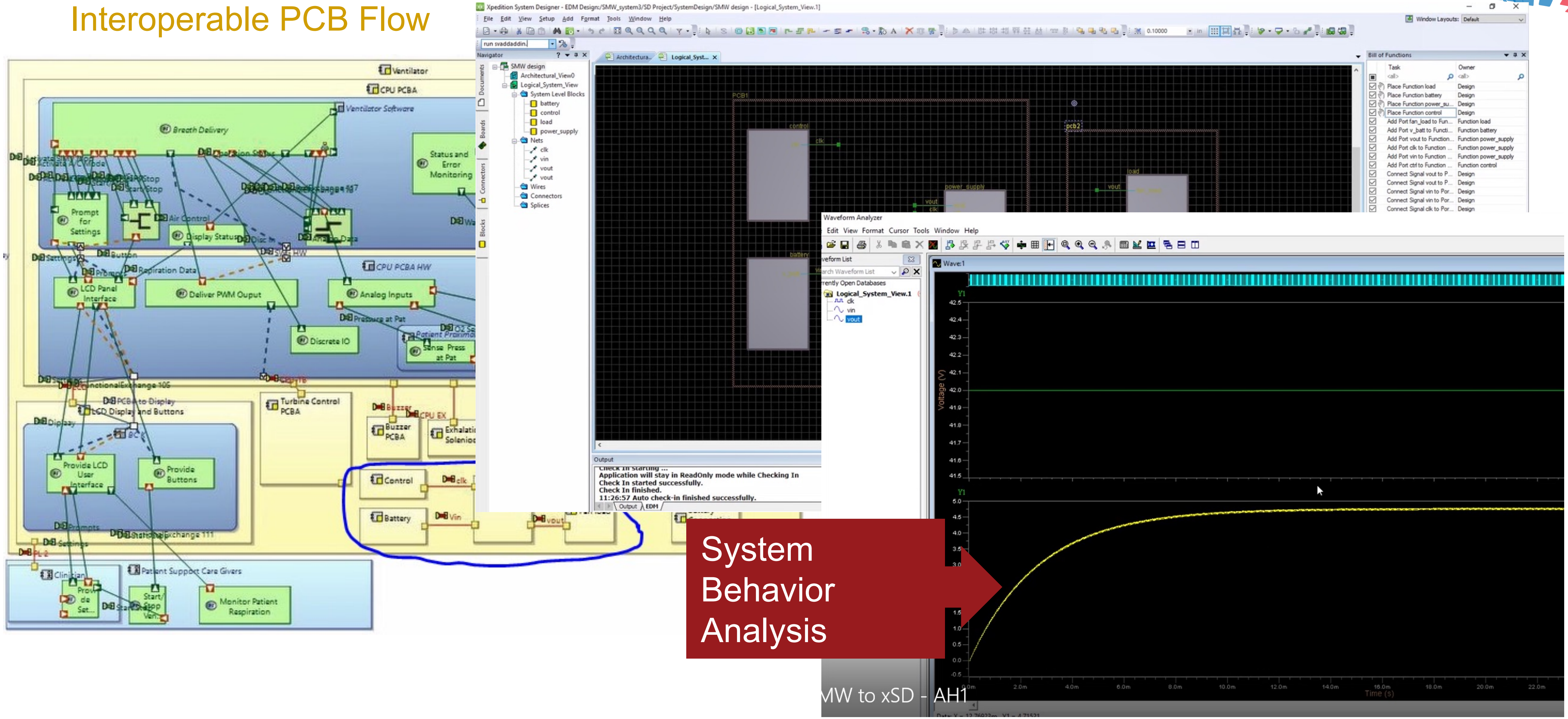
Name	Rev...	Releas...	Description	Source	Usage	Result	Units	Measu...	Goal	Min	Max
Max speed 1.1.1 Speed	A			Speed	Output	Pass		72	60	10	75
Network bandwidth 1.1.4 B...	A			Bandwidth	Output	Fail		180	65	20	80
Time to object 1.1.2 Time t...	A			Time to object detection	Output		sec		80	10	100
VnVParaDefDouble 1.1.2 Ti...	A			Time to object detection	Output				50	40	100



Name	Rev...	Releas...	Description	Source	Usage	Result	Units	Measu...	Goal	Min	Max
Max speed 1.1.1 Speed	A			Speed	Output	Pass		72	60	10	75
Network bandwidth 1.1.4 B...	A			Bandwidth	Output	Pass		60	65	20	80
Time to object 1.1.2 Time t...	A			Time to object detection	Output		sec		80	10	100
VnVParaDefDouble 1.1.2 Ti...	A			Time to object detection	Output				50	40	100

Architecture Implementation; Interoperable PCB Flow

PCB Level Analysis



Bill of Functions

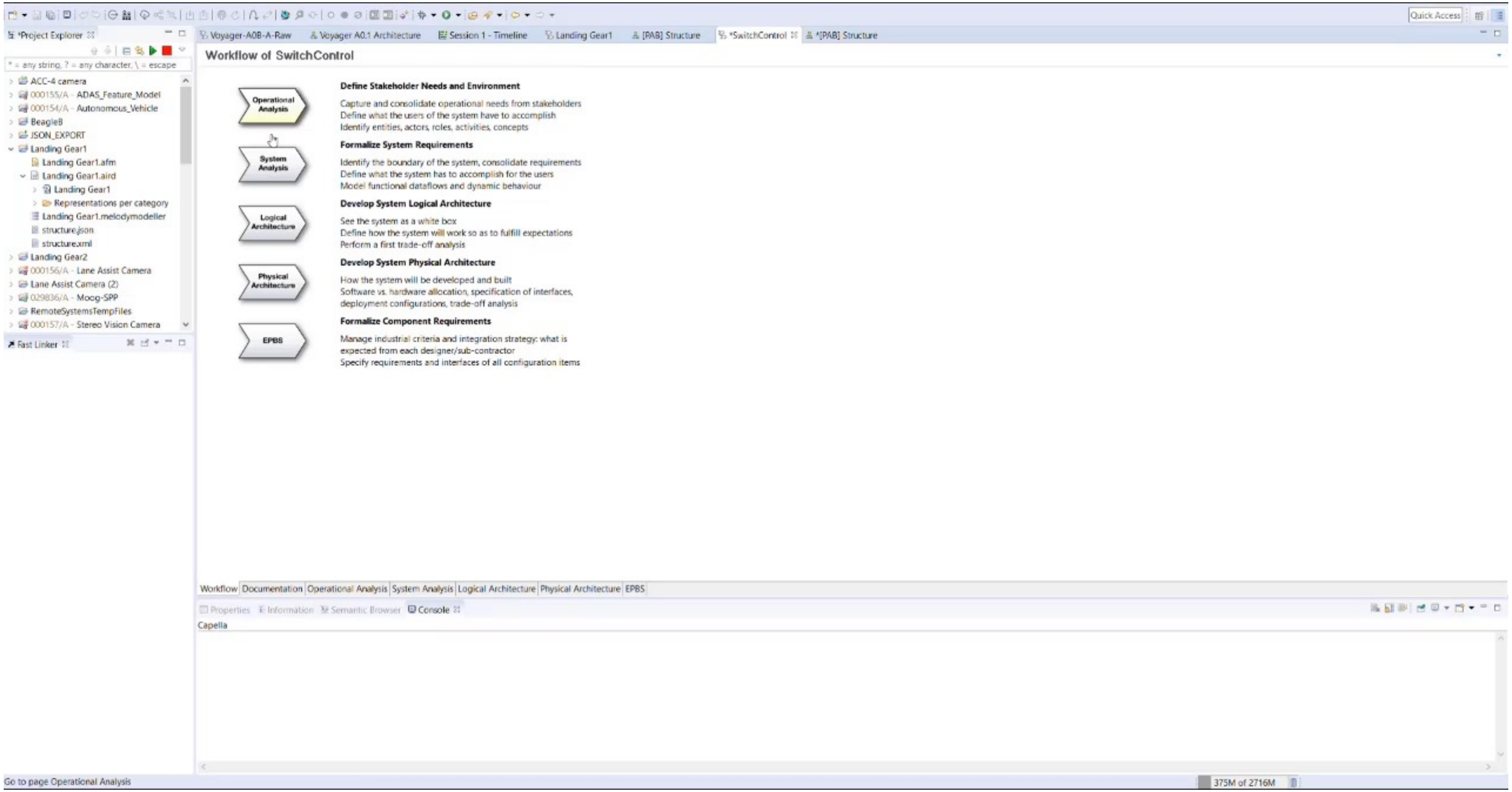
	Task	Owner
■	<all>	<all>
☑	Place Function load	Design
☑	Place Function battery	Design
☑	Place Function power_su...	Design
☑	Place Function control	Design
☑	Add Port fan_load to Fun...	Function load
☑	Add Port v_batt to Functi...	Function battery
☑	Add Port vout to Function...	Function power_supply
☑	Add Port clk to Function ...	Function power_supply
☑	Add Port vin to Function ...	Function power_supply
☑	Add Port ctrl to Function ...	Function control
☑	Connect Signal vout to P...	Design
☑	Connect Signal vout to P...	Design
☑	Connect Signal vin to Por...	Design
☑	Connect Signal vin to Por...	Design
☑	Connect Signal clk to Por...	Design

Output

```
Check In starting ...
Application will stay in ReadOnly mode while Checking In
Check In started successfully.
Check In finished.
11:26:57 Auto check-in finished successfully.
```

System Behavior Analysis

PCB Design Example



Workflow of SwitchControl

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

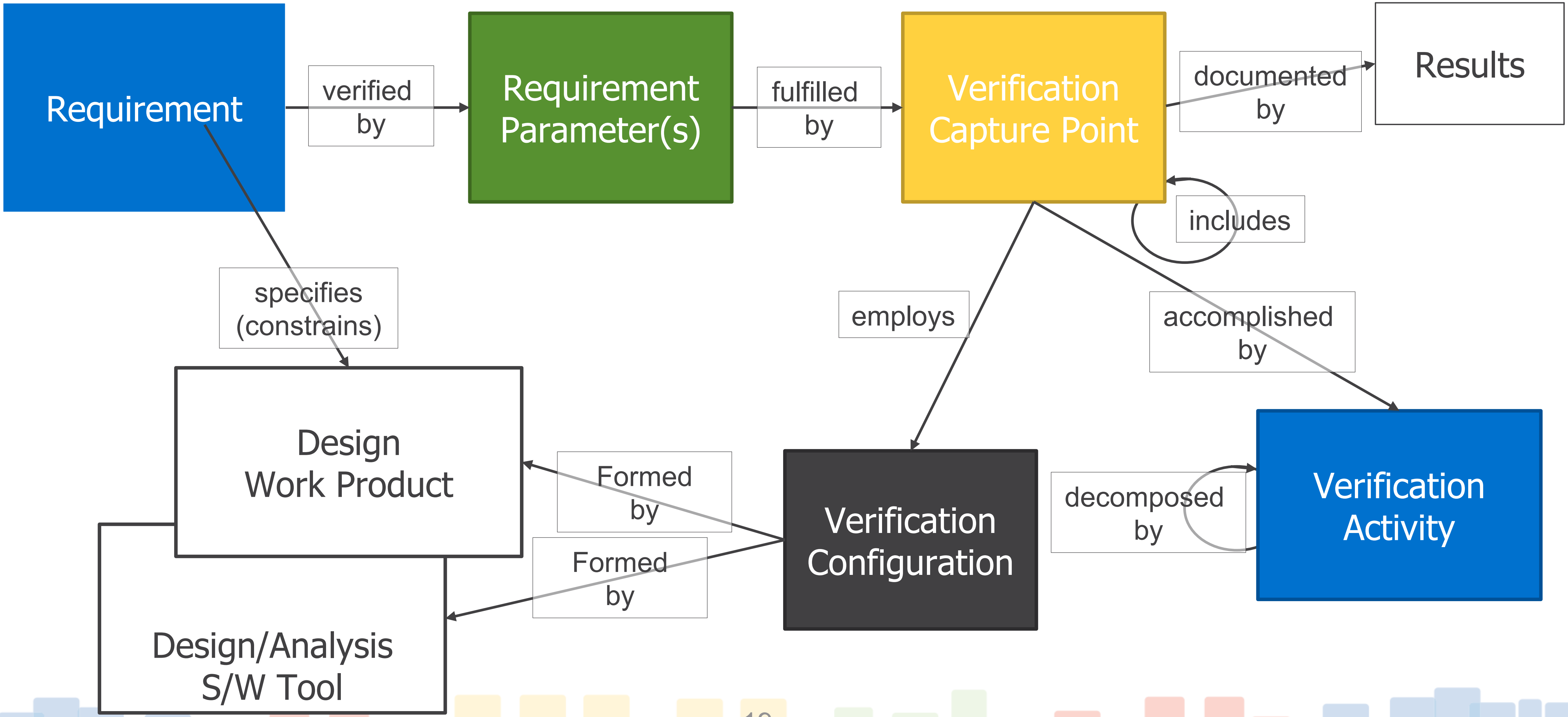
Formalize Component Requirements

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

Go to page Operational Analysis

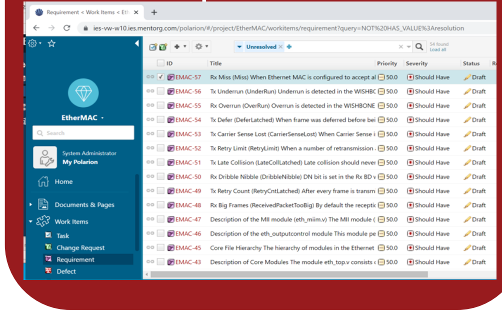
How Do You Digitally Thread Design Verification?

Verification Capture Point (VCP) Metamodel



Creating a Verification Thread

Requirements



Requirement < Work Items < Eth

ies-vw-w10.ies.mentorg.com/polarion/#/project/EtherMAC/workitems/requirement?query=NOT%20HAS_VALUE%3Aresolution

EtherMAC

System Administrator
My Polarion

- Home
- Documents & Pages
- Work Items
  - Task
  - Change Request
  - Requirement
  - Defect

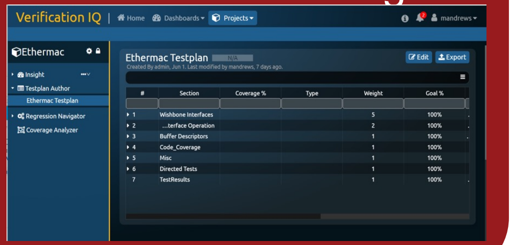
Unresolved — 54 found, Load all

ID	Title	Priority	Severity	Status
EMAC-57	Rx Miss (Miss) When Ethernet MAC is configured to accept al	50.0	Should Have	Draft
EMAC-56	Tx Underrun (UnderRun) Underrun is detected in the WISHBC	50.0	Should Have	Draft
EMAC-55	Rx Overrun (OverRun) Overrun is detected in the WISHBONE	50.0	Should Have	Draft
EMAC-54	Tx Defer (DeferLatched) When frame was deferred before bei	50.0	Should Have	Draft
EMAC-53	Tx Carrier Sense Lost (CarrierSenseLost) When Carrier Sense i	50.0	Should Have	Draft
EMAC-52	Tx Retry Limit (RetryLimit) When a number of retransmission	50.0	Should Have	Draft
EMAC-51	Tx Late Collision (LateCollLatched) Late collision should never	50.0	Should Have	Draft
EMAC-50	Rx Dribble Nibble (DribbleNibble) DN bit is set in the Rx BD v	50.0	Should Have	Draft
EMAC-49	Tx Retry Count (RetryCntLatched) After every frame is transm	50.0	Should Have	Draft
EMAC-48	Rx Big Frames (ReceivedPacketTooBig) By default the receptic	50.0	Should Have	Draft
EMAC-47	Description of the MII module (eth_miim.v) The MII module (	50.0	Should Have	Draft
EMAC-46	Description of the eth_outputcontrol module This module pe	50.0	Should Have	Draft
EMAC-45	Core File Hierarchy The hierarchy of modules in the Ethernet	50.0	Should Have	Draft
EMAC-43	Description of Core Modules The module eth_top.v consists	50.0	Should Have	Draft

OSLC

OSLC

Verification Manager



Verification IQ | Home | Dashboards | Projects | mandrews

Ethermac

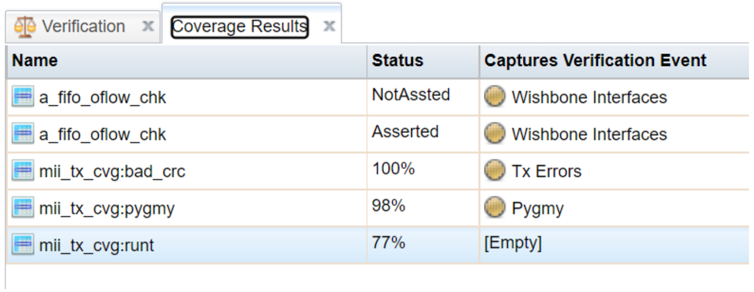
- Insight
- Testplan Author
  - Ethermac Testplan
- Regression Navigator
- Coverage Analyzer

Ethermac Testplan N/A — Edit — Export

Created By admin, Jun 1. Last modified by mandrews, 7 days ago.

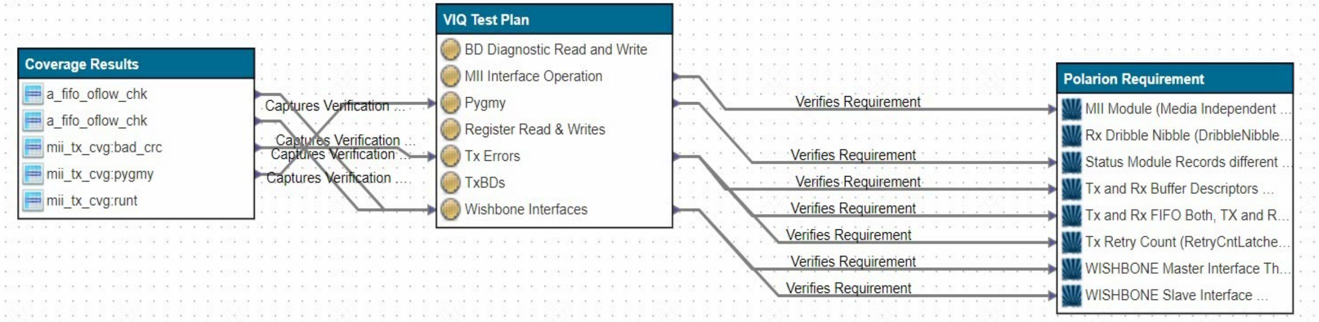
#	Section	Coverage %	Type	Weight	Goal %
1	Wishbone Interfaces			5	100%
2	...terface Operation			2	100%
3	Buffer Descriptors			1	100%
4	Code_Coverage			1	100%
5	Misc			1	100%
6	Directed Tests			1	100%
7	TestResults			1	100%

Electronics Verification Thread



Verification — Coverage Results

Name	Status	Captures Verification Event
a_fifo_oflow_chk	NotAssted	Wishbone Interfaces
a_fifo_oflow_chk	Asserted	Wishbone Interfaces
mii_tx_cvg:bad_crc	100%	Tx Errors
mii_tx_cvg:pygmy	98%	Pygmy
mii_tx_cvg:runt	77%	[Empty]

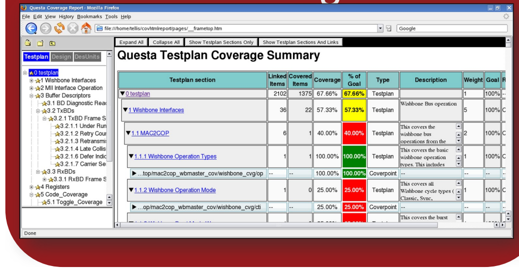


Coverage Results
- a_fifo_oflow_chk
- a_fifo_oflow_chk
- mii_tx_cvg:bad_crc
- mii_tx_cvg:pygmy
- mii_tx_cvg:runt

VIQ Test Plan
- BD Diagnostic Read and Write
- MII Interface Operation
- Pygmy
- Register Read & Writes
- Tx Errors
- TxBDs
- Wishbone Interfaces

Polarion Requirement
- MII Module (Media Independent ...
- Rx Dribble Nibble (DribbleNibble...
- Status Module Records different ...
- Tx and Rx Buffer Descriptors ...
- Tx and Rx FIFO Both, TX and R...
- Tx Retry Count (RetryCntLatche...
- WISHBONE Master Interface Th...
- WISHBONE Slave Interface ...

Coverage



Questa Coverage Report - Mozilla Firefox

File Edit View History Bookmarks Tools Help

file:///home/tellis/covhtmlreport/pages/__frametop.htm

Testplan Design DesUnits

- 0 testplan
  - 1 Wishbone Interfaces
  - 2 MII Interface Operation
  - 3 Buffer Descriptors
    - 3.1 BD Diagnostic Reac
    - 3.2 TxBDs
      - 3.2.1 TxBD Frame S
        - 3.2.1.1 Under Run
        - 3.2.1.2 Retry Cour
        - 3.2.1.3 Retransmi
        - 3.2.1.4 Late Collis
        - 3.2.1.6 Defer Indic
        - 3.2.1.7 Carrier Se
    - 3.3 RxBDs
      - 3.3.1 RxBD Frame S
  - 4 Registers
  - 5 Code_Coverage
    - 5.1 Toggle_Coverage

Expand All | Collapse All | Show Testplan Sections Only | Show Testplan Sections And Links

Questa Testplan Coverage Summary

Testplan section	Linked Items	Covered Items	Coverage	% of Goal	Type	Description	Weight	Goal
0 testplan	2102	1375	67.66%	67.66%	Testplan		1	100%
1 Wishbone Interfaces	36	22	57.33%	57.33%	Testplan	Wishbone Bus operation	5	100%
1.1 MAC2COP	6	1	40.00%	40.00%	Testplan	This covers the wishbone bus operations from the	2	100%
1.1.1 Wishbone Operation Types	1	1	100.00%	100.00%	Testplan	This covers the basic wishbone operation types. This includes	1	100%
...top/mac2cop_wbmaster_cov/wishbone_cvg/op	--	--	100.00%	100.00%	Coverpoint	--	--	--
1.1.2 Wishbone Operation Mode	1	0	25.00%	25.00%	Testplan	This covers all Wishbone cycle types (Classic, Sync,	1	100%
...op/mac2cop_wbmaster_cov/wishbone_cvg/cti	--	--	25.00%	25.00%	Coverpoint	--	--	--

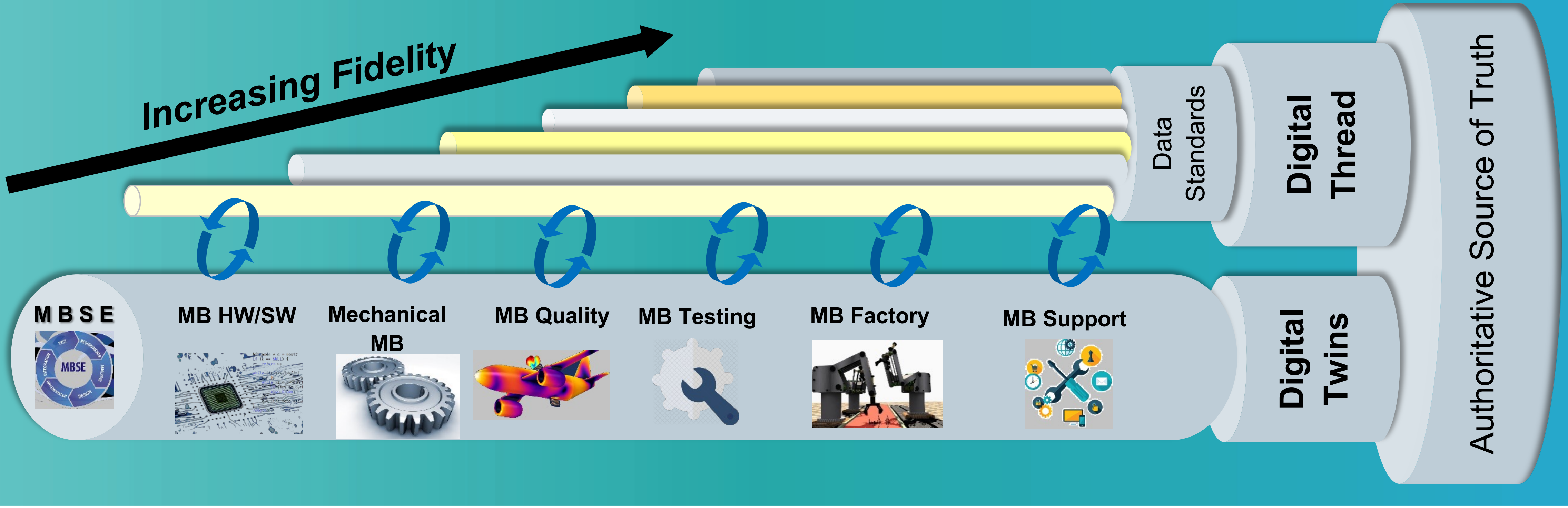
Done

OSLC

Digital Transformation is Enabled by Digital Twinning and Threading

Digital Thread: The authoritative technical data providing decision makers the right data at the right time across the system life cycle

Digital Twin: An integrated digital simulation, enabled by digital threading



Increasing Fidelity

MBSE

MB HW/SW

Mechanical MB

MB Quality

MB Testing

MB Factory

MB Support

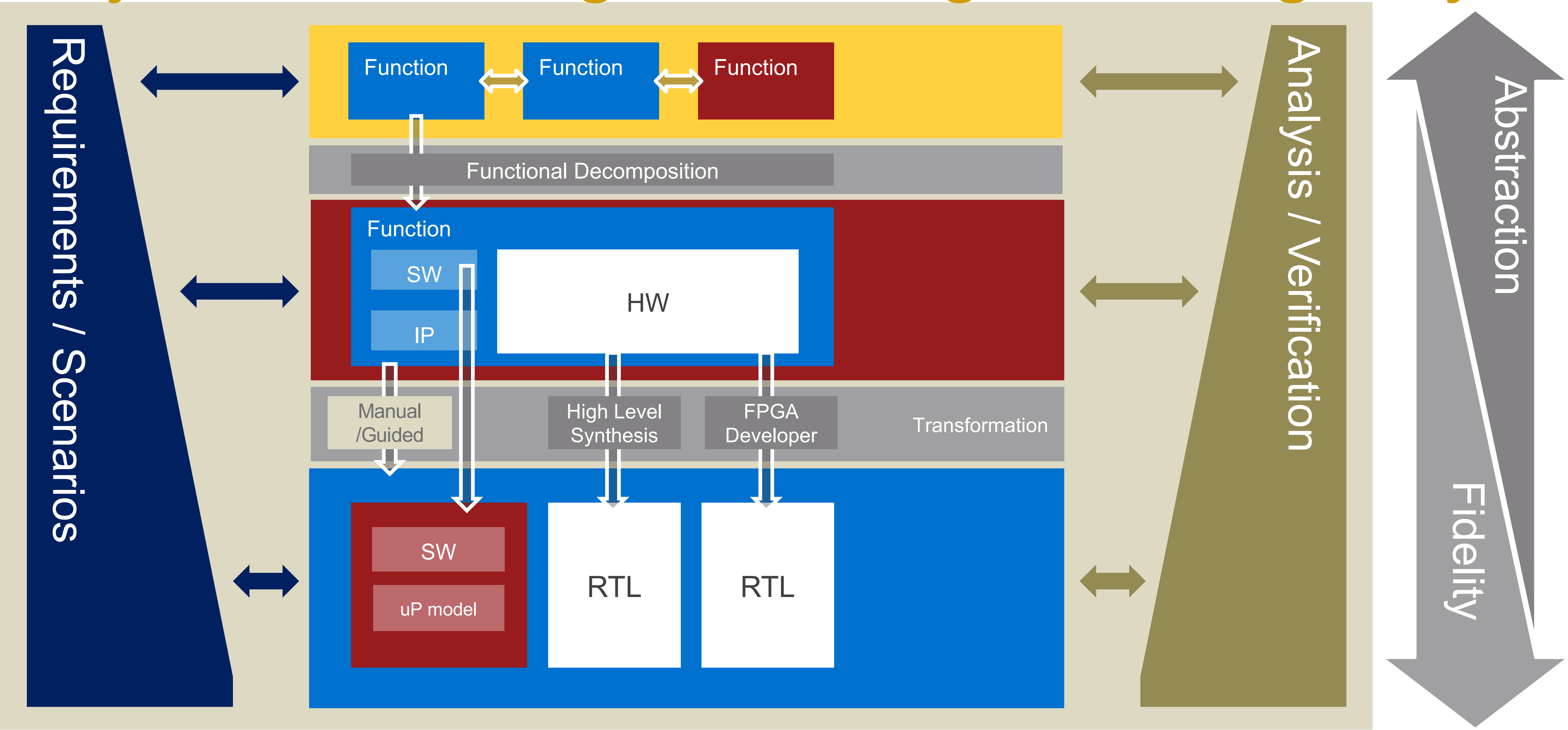
Data Standards

Digital Thread

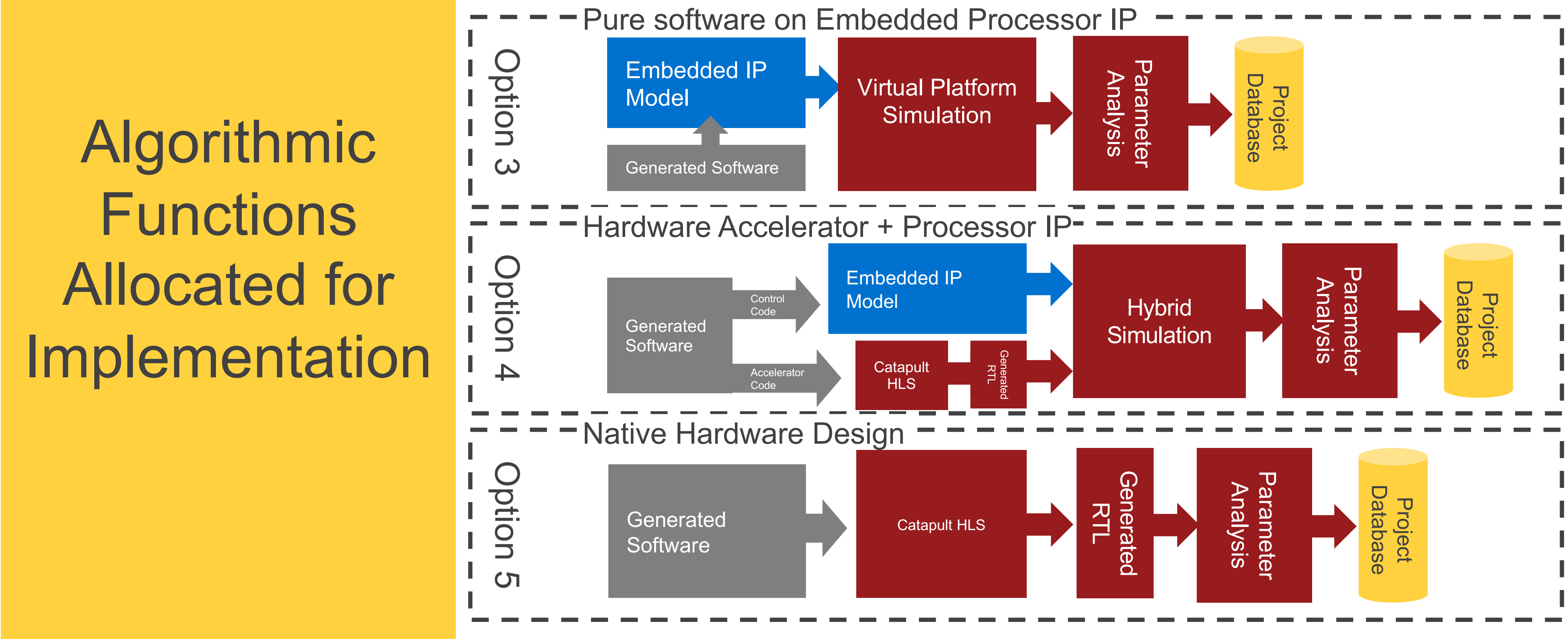
Digital Twins

Authoritative Source of Truth

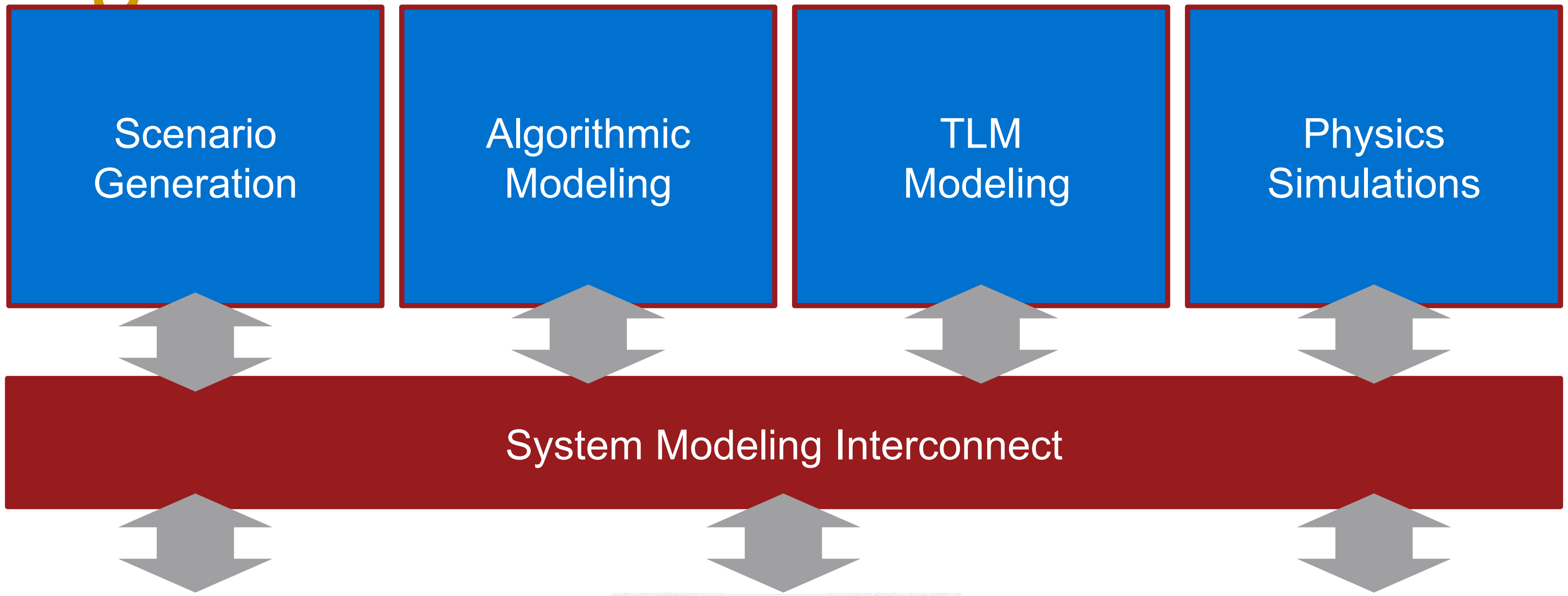
Hybrid Twinning in the Digital Design Cycle



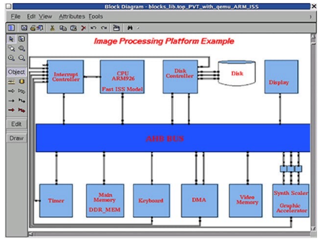
Automated Design Synthesis and Analysis



Future State: Seamless Heterogeneous Modeling Environment



FMI Enabled



Virtual Platform

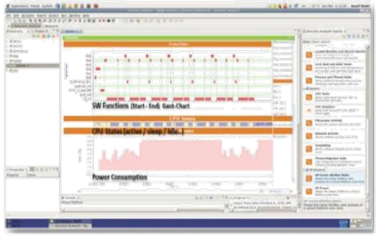

SW Functions (Start - End) Gant-Chart
CPU States (active / sleep / idle..)
Power Consumption

Fig.5 – Logged profile info over time [source: Mentor]

Simulation

Hardware Assisted

Summary and Next Steps

- Platforms for digitally threaded design and verification of electronics are being piloted.
- Initial results are encouraging, as traditionally siloed tools show seamless interoperable potential.
- Standards adoption (e.g. SysML V2, FMI, OSLC) offers promise to realize expansion into multi-vendor DTDV platforms.

32nd Annual INCOSE
international symposium

hybrid event

Detroit, MI, USA
June 25 - 30, 2022

www.incose.org/symp2022