30 June 2020

AUTOSAR Architecture Modeling of Multi-core Electric Powertrain Controller

Dr Sakthivel Manikandan Sundharam /
Software Architect

Delphi Technologies

Bio : Sakthivel Manikandan Sundharam

True! Bit longer name - Shortly “Sakthi”

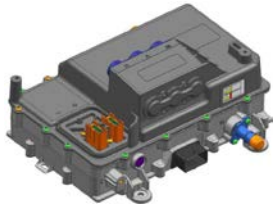
- Software Architect – Powertrain Electrification & Electronics
- 15+ Years of Automotive Embedded System Experience
- Ph.D. in Timing Aware Model-Based Design to Automotive Embedded Systems, University of Luxembourg, Luxembourg
- Masters in Embedded Systems, College of Engineering Chennai, India
- Work revolves around software architectural topics incorporating timing, memory, and safety constraints of automotive software.

Outline / Agenda

1. Delphi Technologies - Powertrain Electrification Product Portfolio
2. HV Inverter System Context
3. Pitfalls in Legacy Approach of SW Architecture Modeling
4. Evaluation of Journey
   - Requirements to Architecture
   - Architecture authoring
   - Interfaces / Data dictionary
5. Lessons learnt and Best practices

Delphi Technologies - Powertrain Electrification Product Portfolio

Low cost, high density, rugged with various levels of integration available



Single Inverter

Inverter w/ DC/DC (CIDD)

48V Inverter for BSG

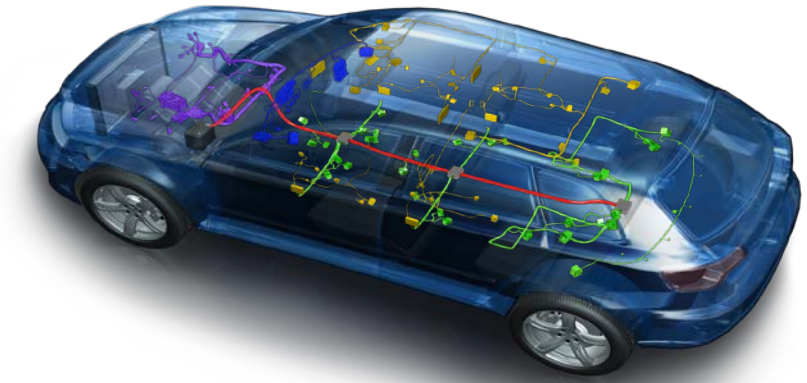
48V DC/DC Converter

High Voltage Box

3-in-1 Inverter

DC/DC Converters

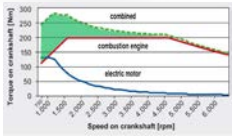
Dual Inverter



Battery Pack Controller

Dual Inverter/ Converter/ Hybrid Controller

Supervisory Controller (Hybrid Control Unit)



Hybrid Control Software

On-board Battery Chargers

Delphi Technologies Inverter – The Next Generation

Inverter with conventional Power Module

Conventional
Many, many wire-bonds

Gravimetric power density (kVA/Kg)

16.1 → 20.6
25% higher

Volume (L)

10.7 → 7.6
30% smaller

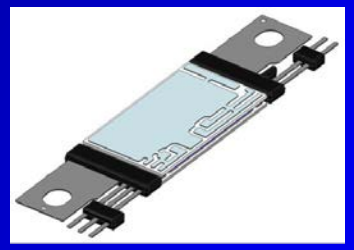
Mass (Kg)

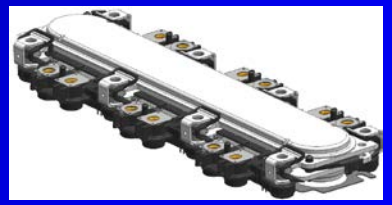
14.5 → 8.4
40% lighter

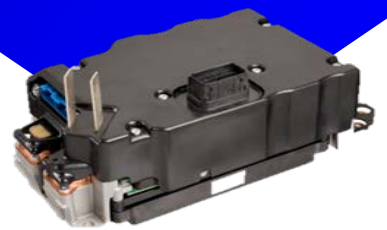
Efficiency Improvement (MPG)

0 → 2

Delphi Technologies Inverter with viper



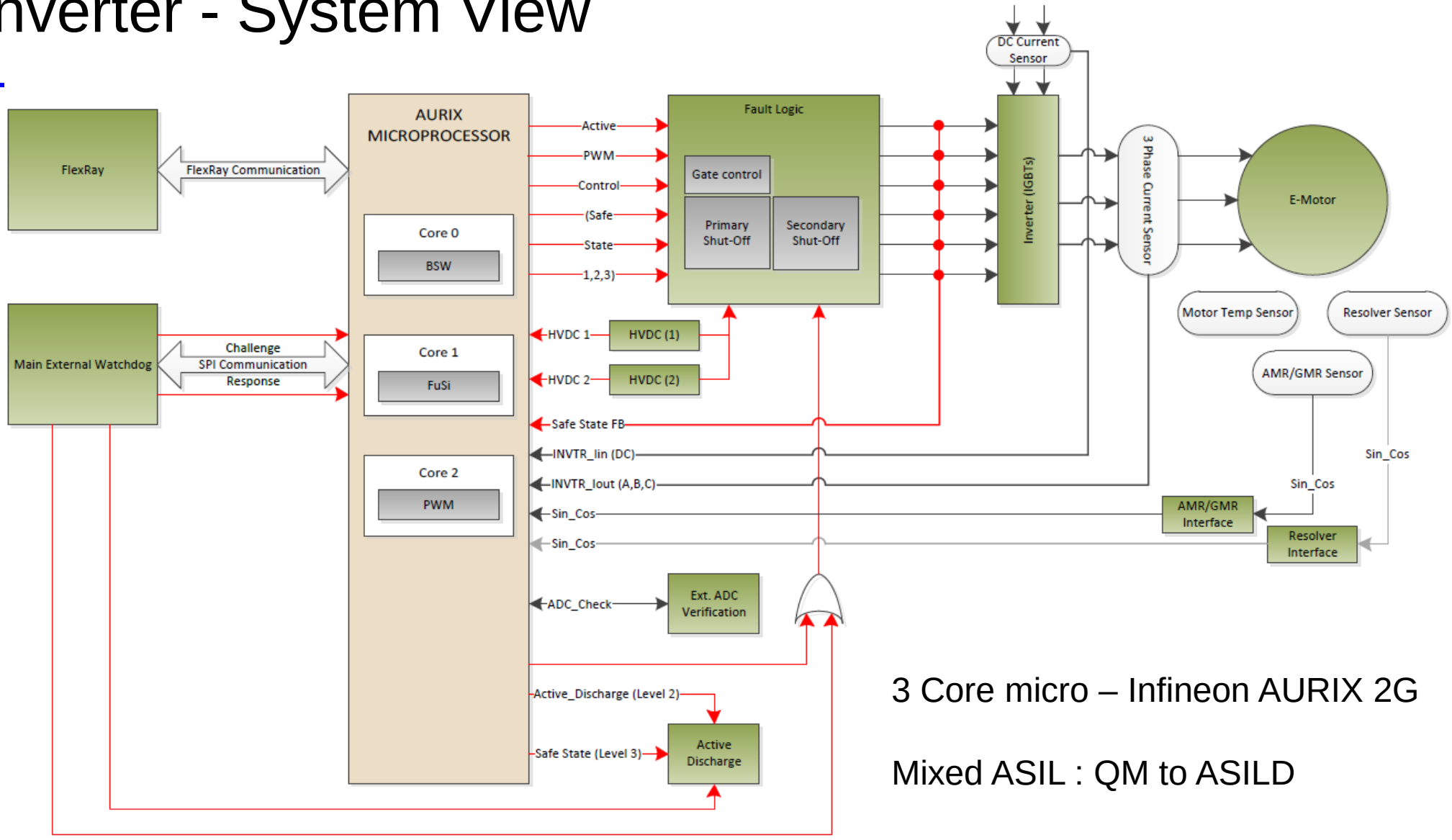




Delphi Inverter next generation

- Next gen Viper enables extra high voltage 800V bus inverters
- Flexibility to move from Si to SiC power switch to enable higher efficiency & lower cost
- Advanced capacitor enables up to 70% reduction in component volume & weight

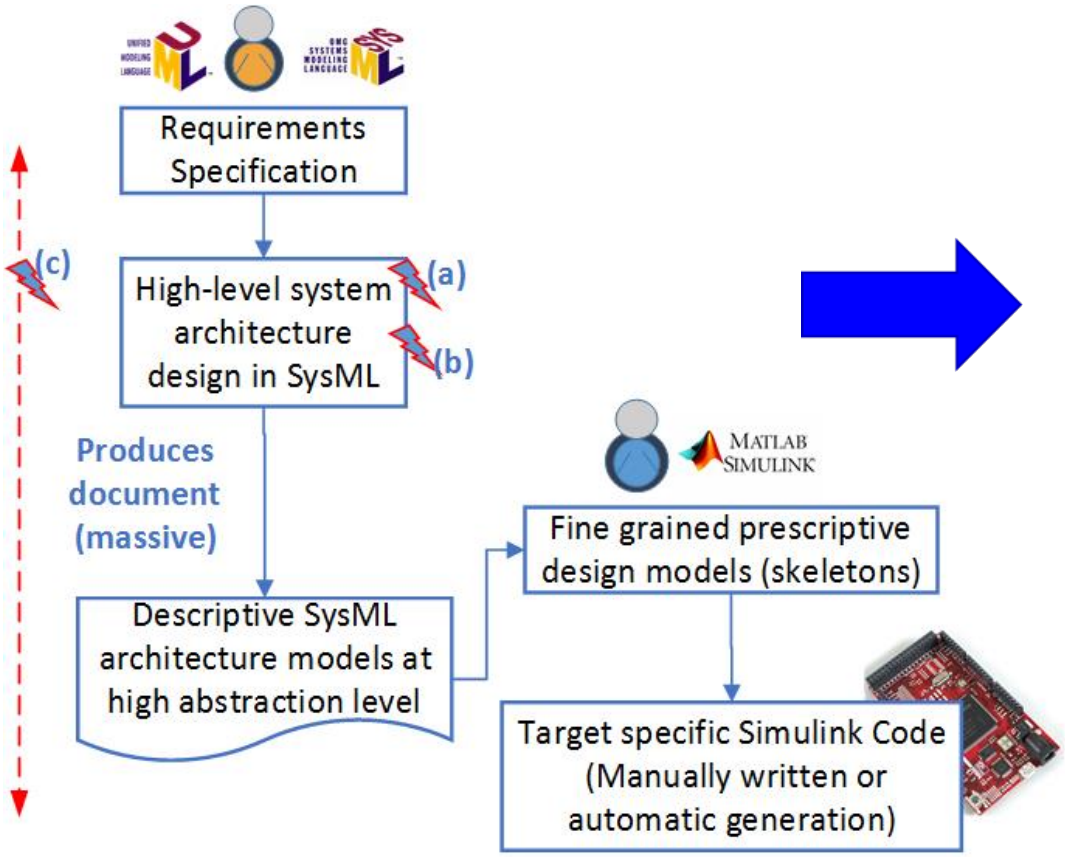
HV Inverter - System View



3 Core micro – Infineon AURIX 2G

Mixed ASIL : QM to ASILD

Pitfalls in Legacy Approach and Best Practices Evaluated

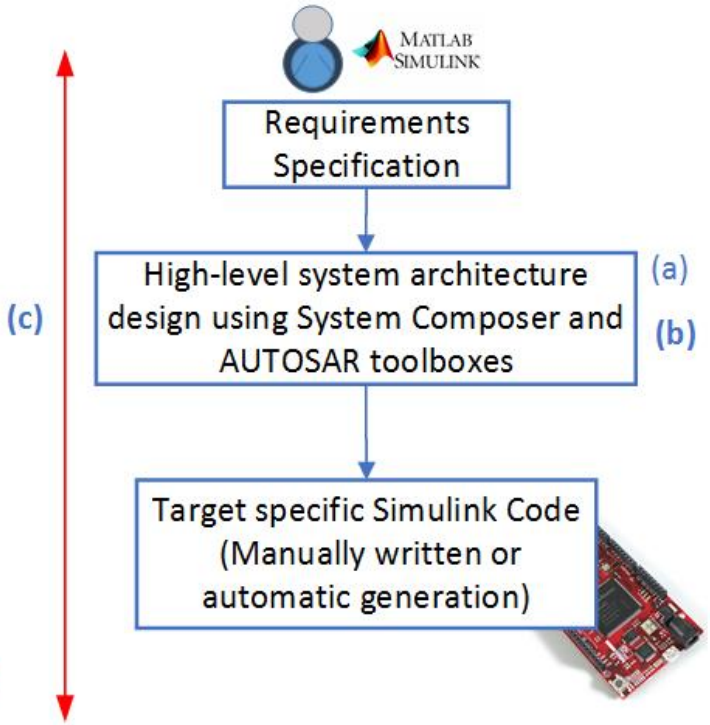


(a) Dearth of tool support for fine-grained modeling of AUTOSAR-based architecture design

(b) Lack of support for proof-of-concept study of what-if scenarios, performance analysis

(c)Gaps in requirements traceability

(1)



(a) AUTOSAR and Simulink System Composer toolboxes for AUTOSAR-based systems engineering

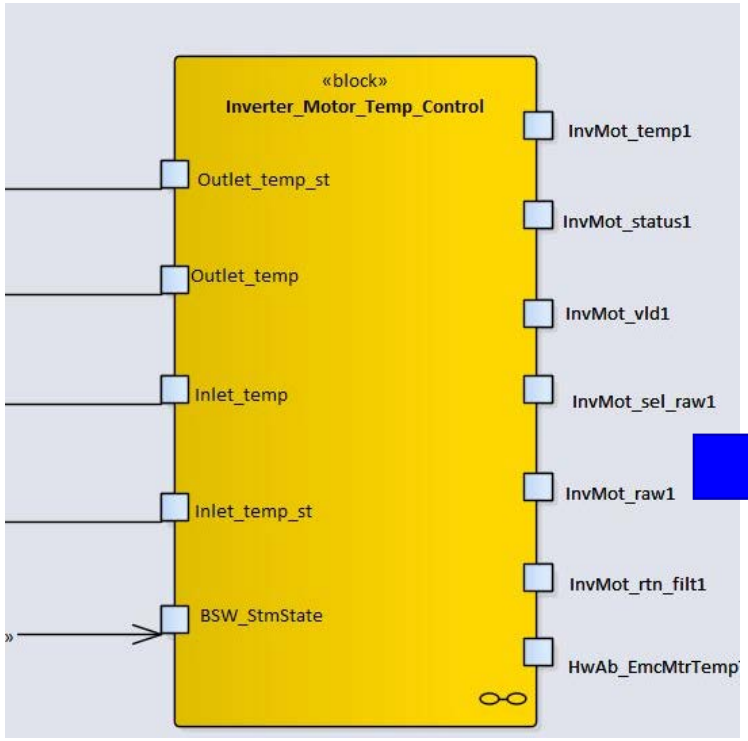
(b)Proof-of-concept studies, performance and viewpoint analysis of various stakeholders

(c)Bidirectional traceability

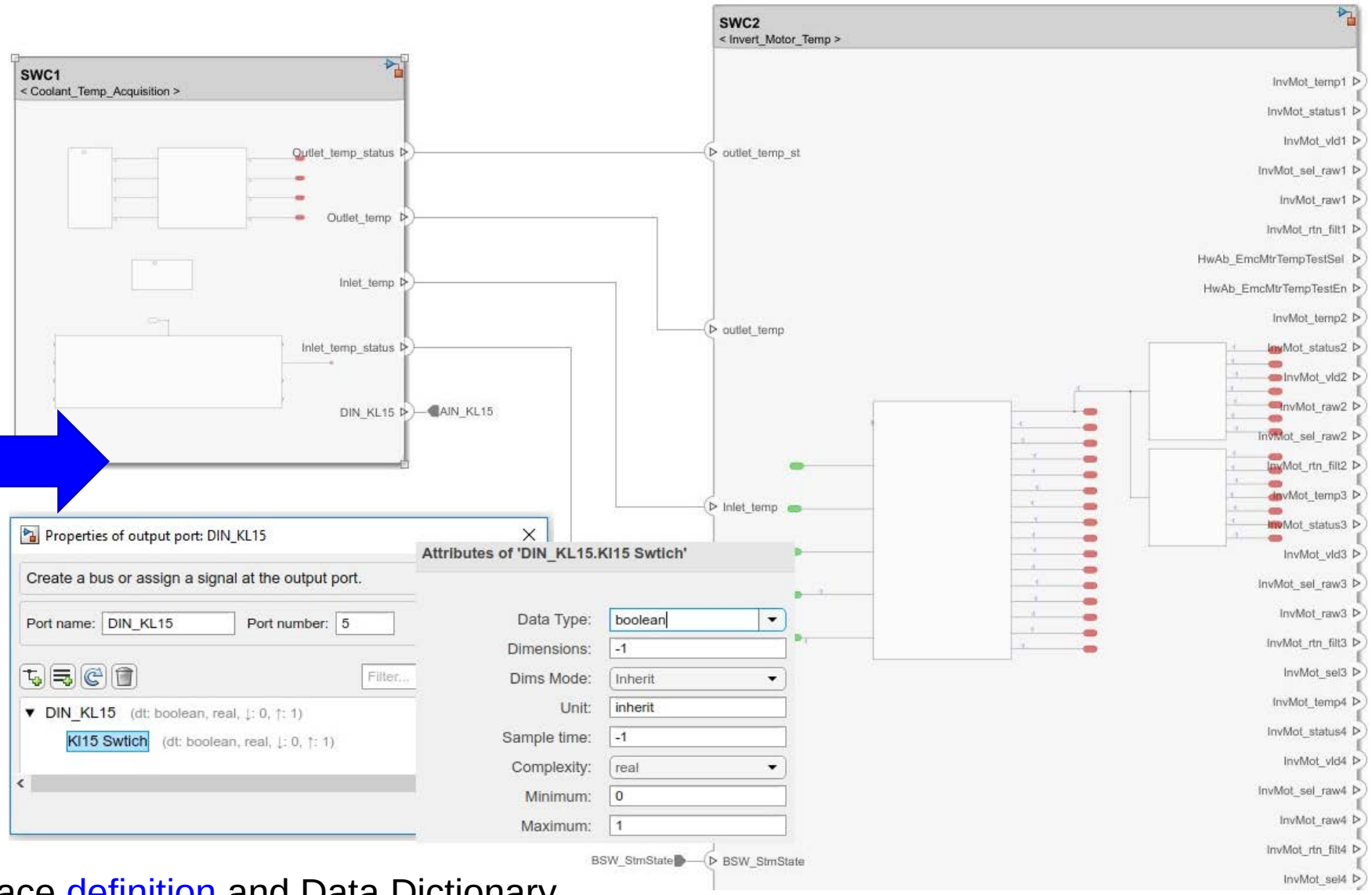
(2)

Static Software Architecture Tooling Twins

MLSL's AUTOSAR Blockset + System Composer



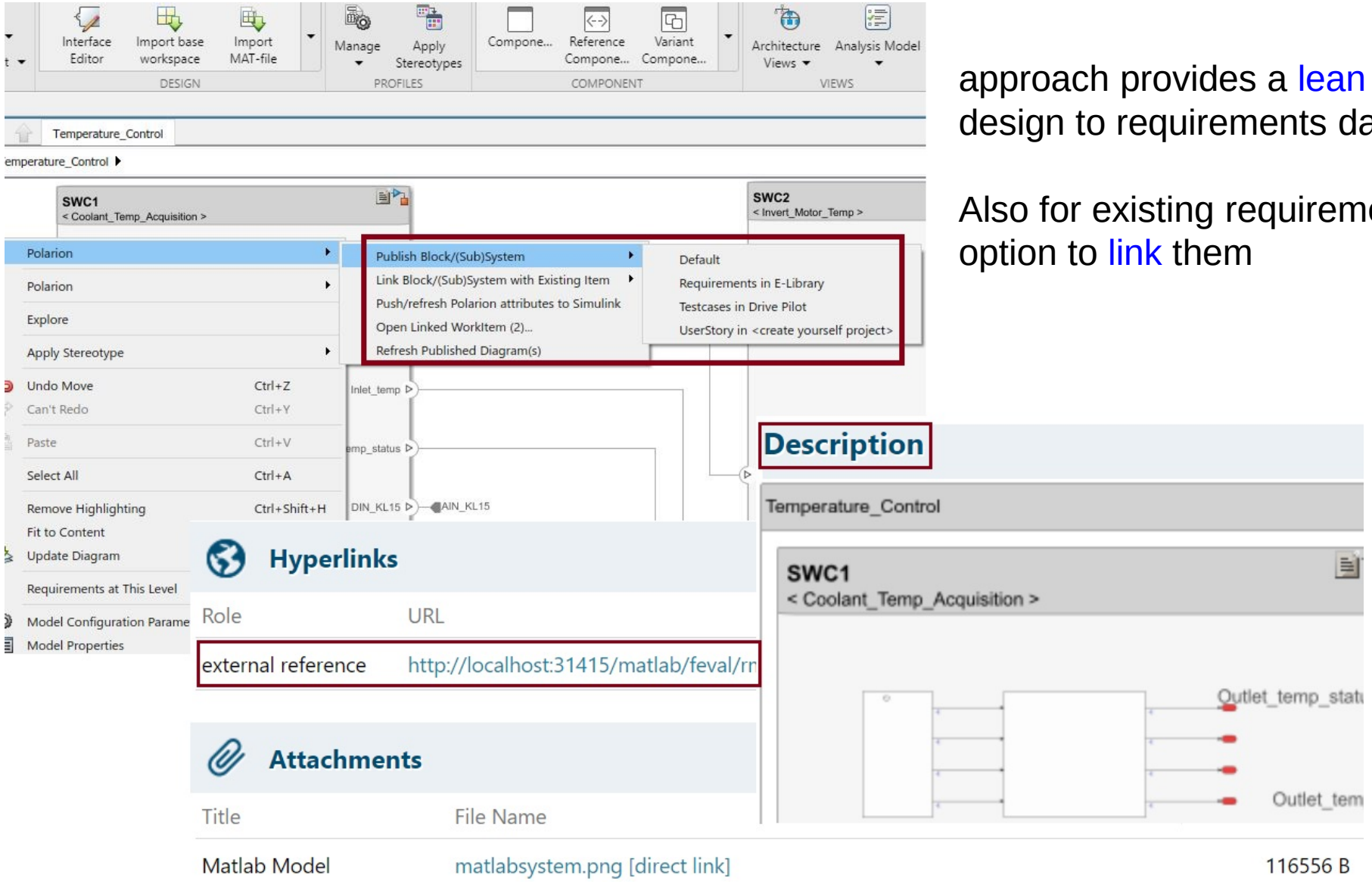
Legacy SysML



Interface definition and Data Dictionary

Delphi Technologies

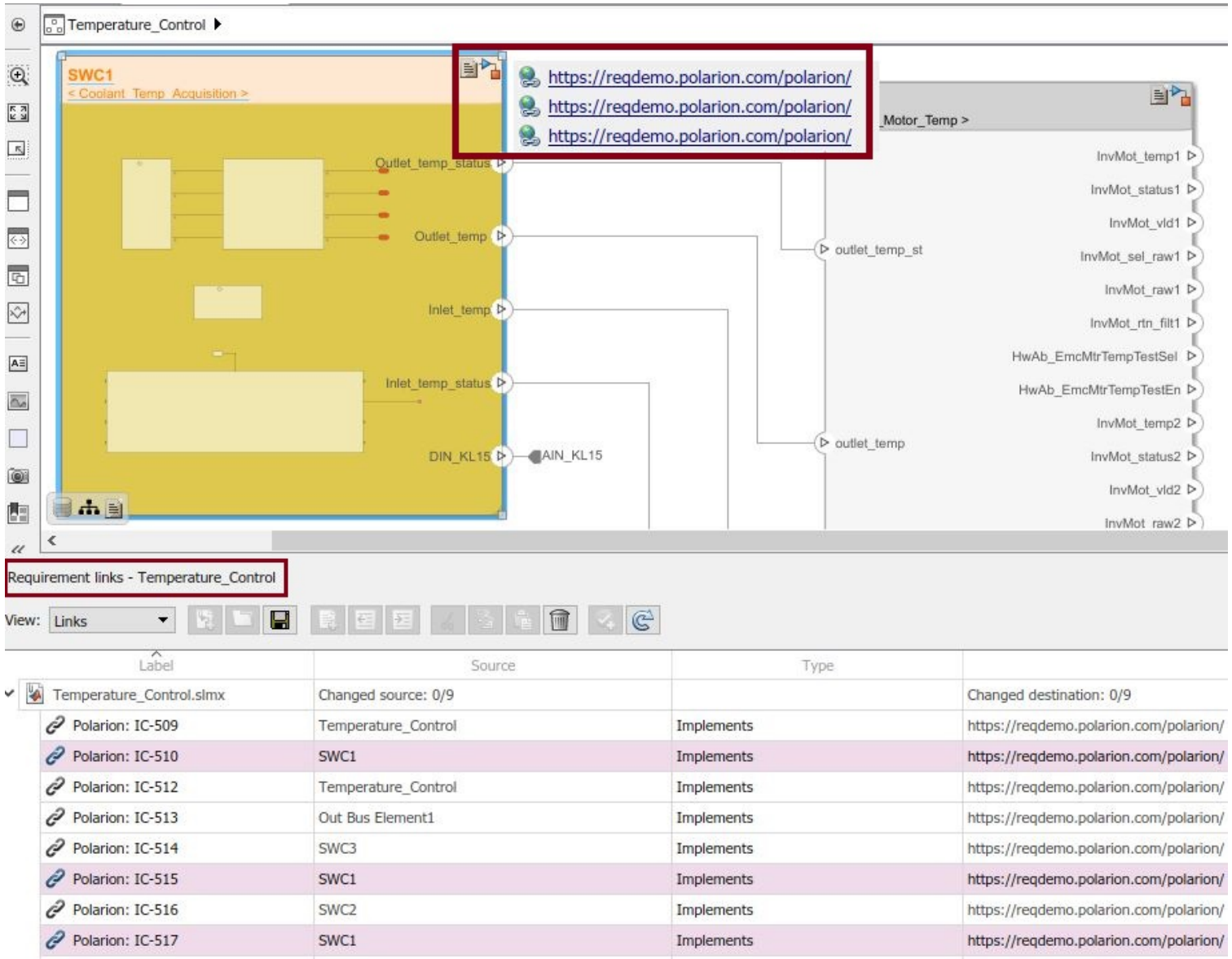
Publishing Architecture Modeling onto Requirements Database



approach provides a lean way to publish the design to requirements database.

Also for existing requirements, it provides an option to link them

Requirements to Architecture Linking



Label	Source	Type	
Temperature_Control.slmx	Changed source: 0/9		Changed destination: 0/9
Polarion: IC-509	Temperature_Control	Implements	https://reqdemo.polarion.com/polarion/
Polarion: IC-510	SWC1	Implements	https://reqdemo.polarion.com/polarion/
Polarion: IC-512	Temperature_Control	Implements	https://reqdemo.polarion.com/polarion/
Polarion: IC-513	Out Bus Element1	Implements	https://reqdemo.polarion.com/polarion/
Polarion: IC-514	SWC3	Implements	https://reqdemo.polarion.com/polarion/
Polarion: IC-515	SWC1	Implements	https://reqdemo.polarion.com/polarion/
Polarion: IC-516	SWC2	Implements	https://reqdemo.polarion.com/polarion/
Polarion: IC-517	SWC1	Implements	https://reqdemo.polarion.com/polarion/

Tracking of requirements back and forth between modeling and requirements database to verify fulfillment of requirements

Bi-directional Traceability - Forward

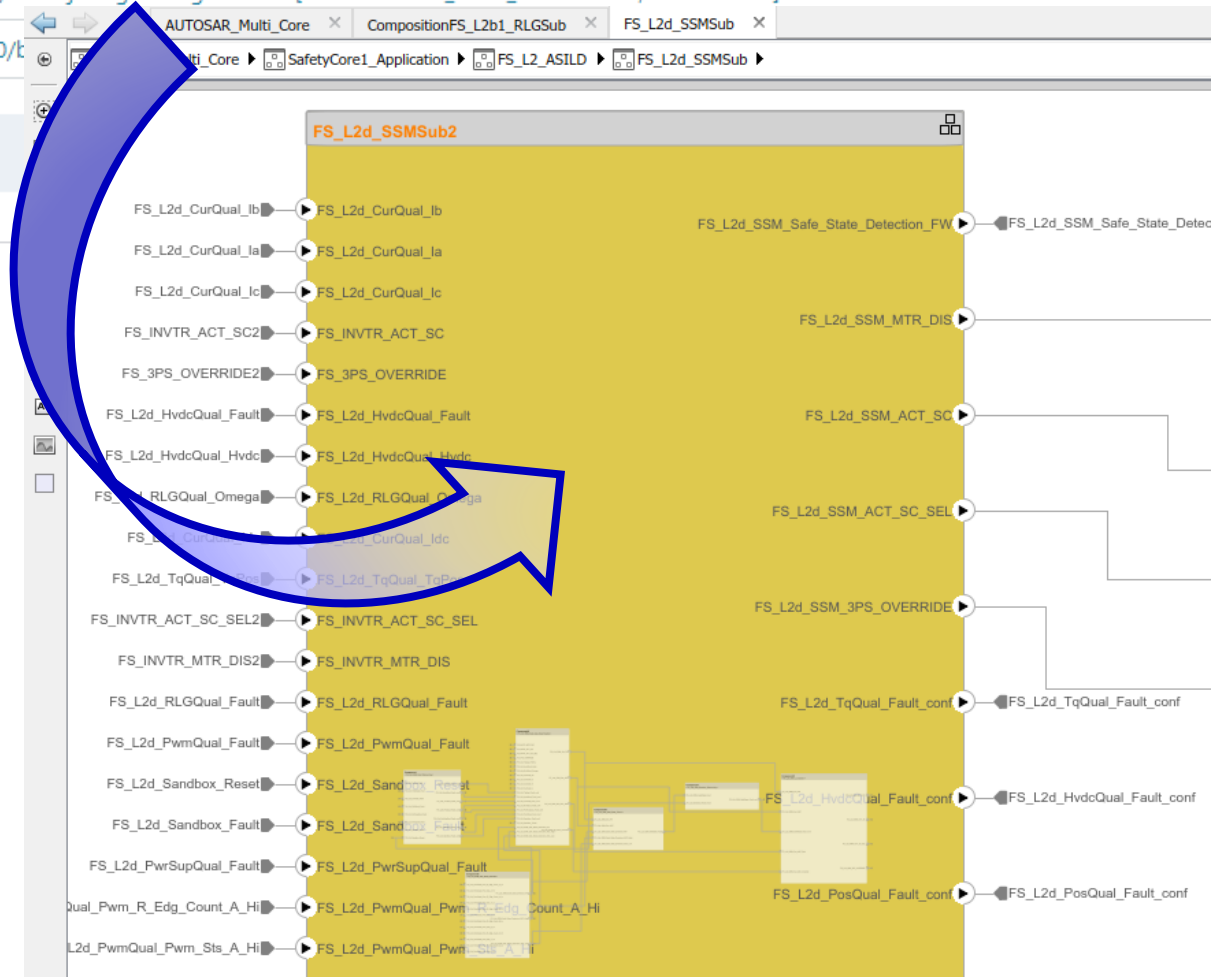
is allocated to AINV-47779 - Reqs - SW Architecture

Hyperlinks

Role	URL
external reference	http://localhost:31415/matlab/feval/rmiobjnavigate?arguments=[%22AUTOSAR_Multi_Core.slx%22,%22:317%22]
external reference	http://jiraprod.delphidrive.com:8080/b

Attachments

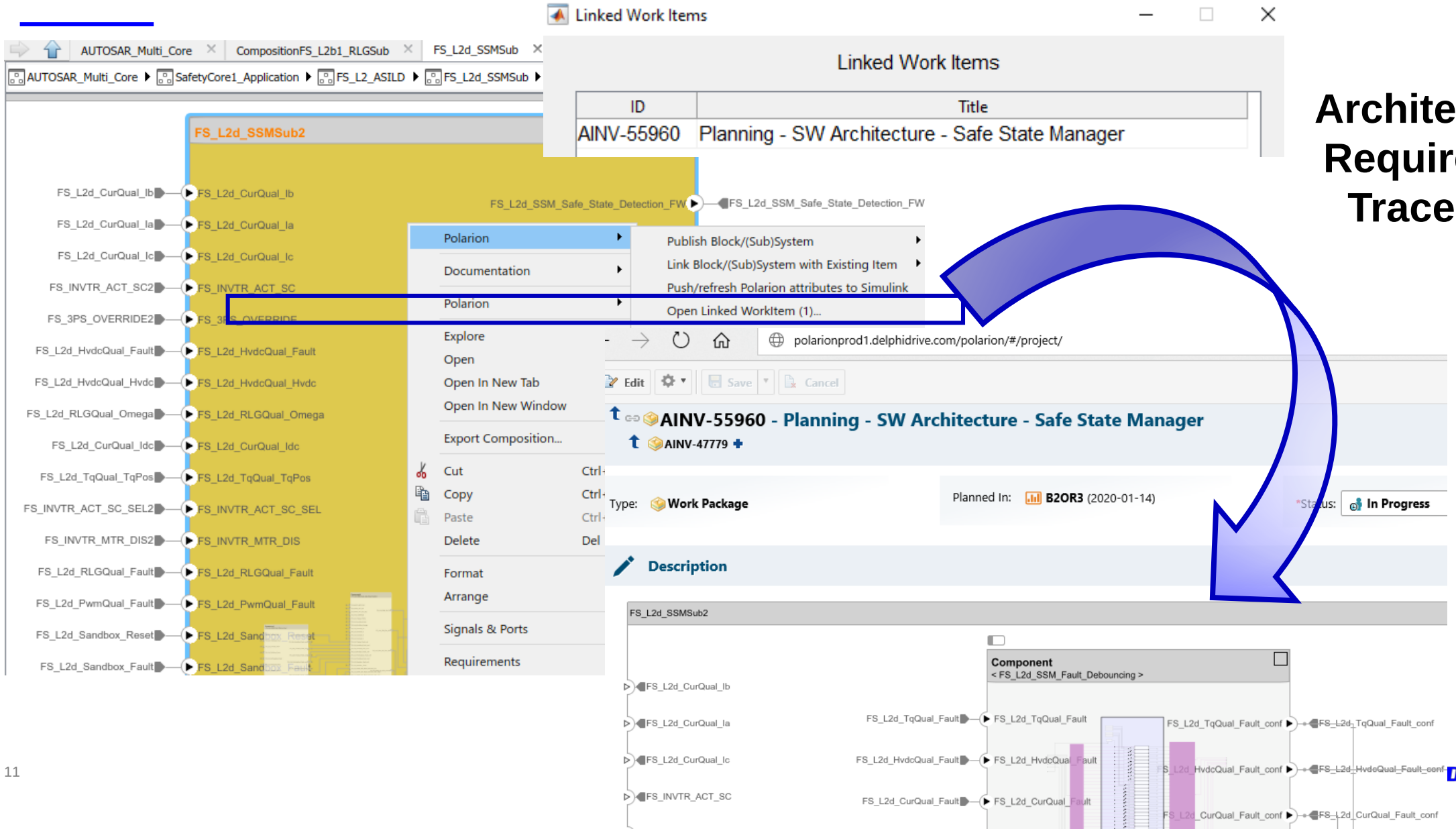
Title	File Name
Matlab Model	matlabsystem.png [direct link]



Requirements to Architecture Traceability

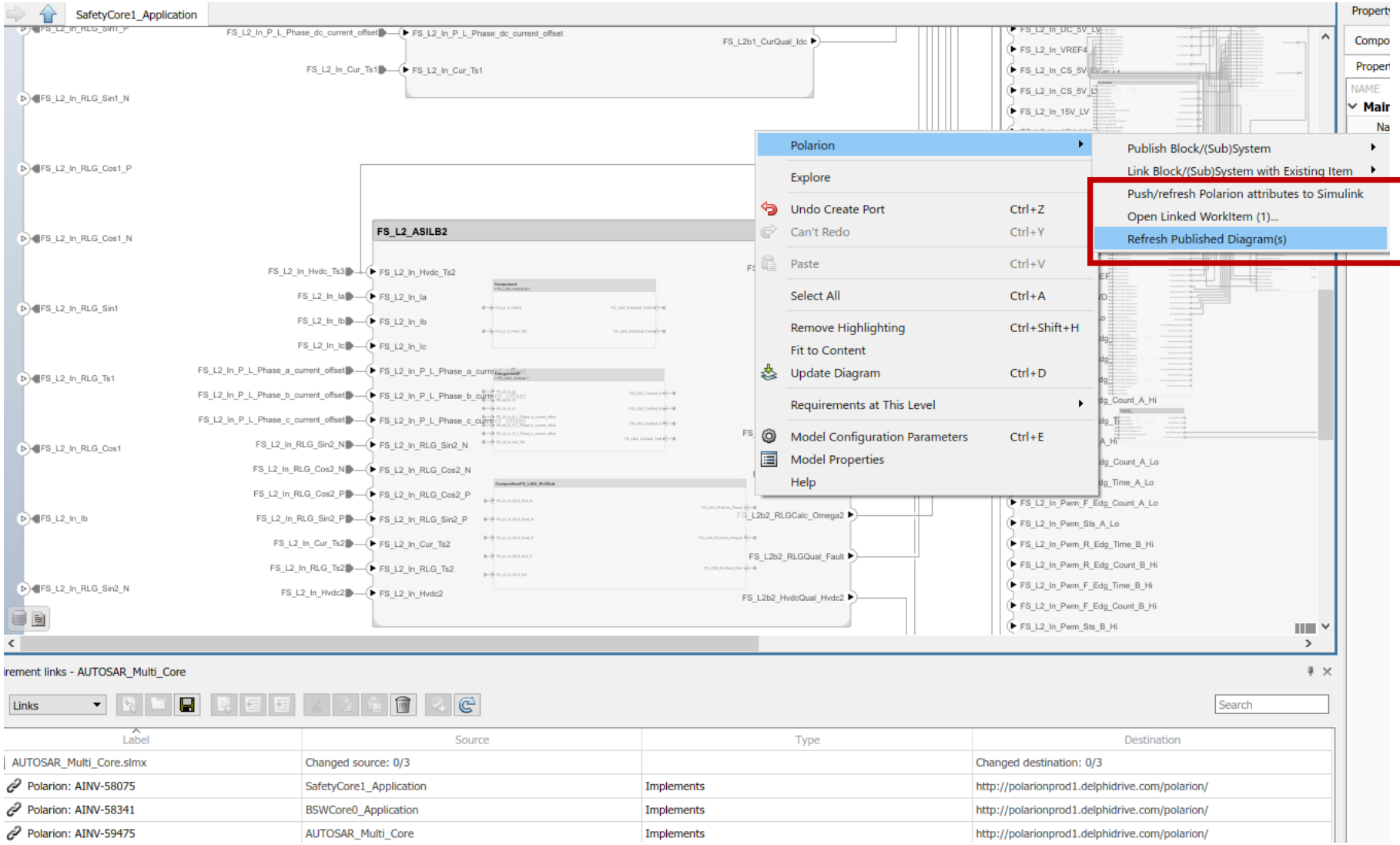
Delphi Technologies

Bi-directional Traceability - Backward



Architecture to Requirements Traceability

Architecture to Requirements – Seamless Approach

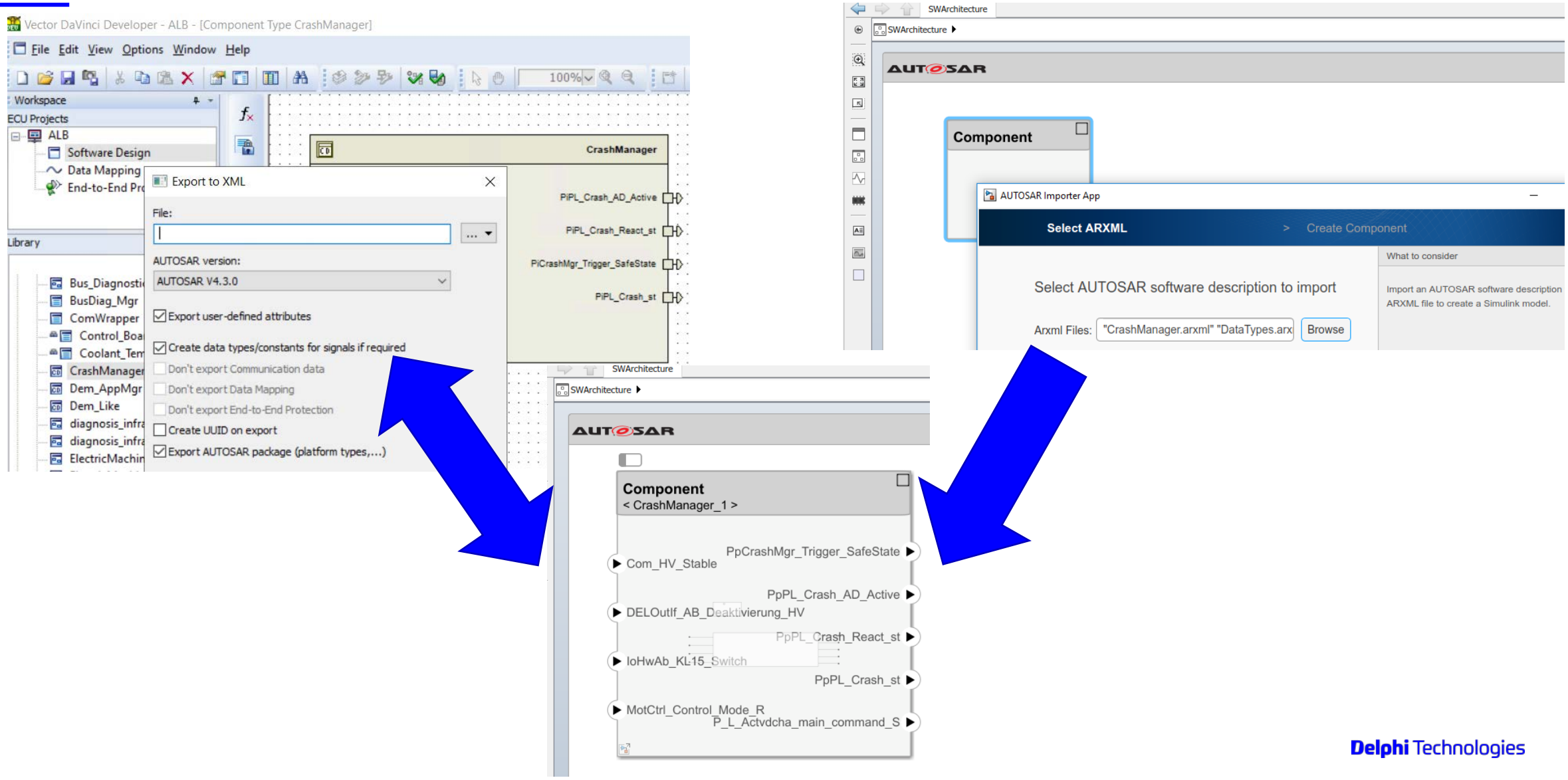


- Whenever model updated due to maturity of the project, refresh option updates the same model onto requirements database

- Reversely, requirement attributes changed on the requirements database can easily be pushed back to SW architecture

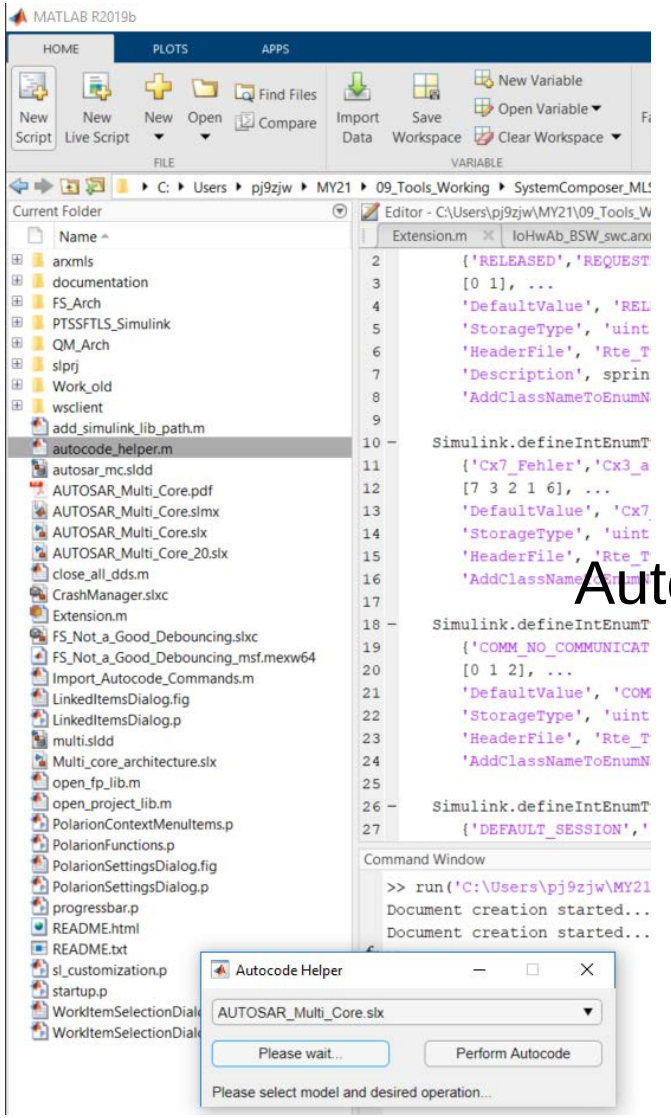
Delphi Technologies

arxml Import from BSW Tools (f.e Vector BSW-stack Tools)

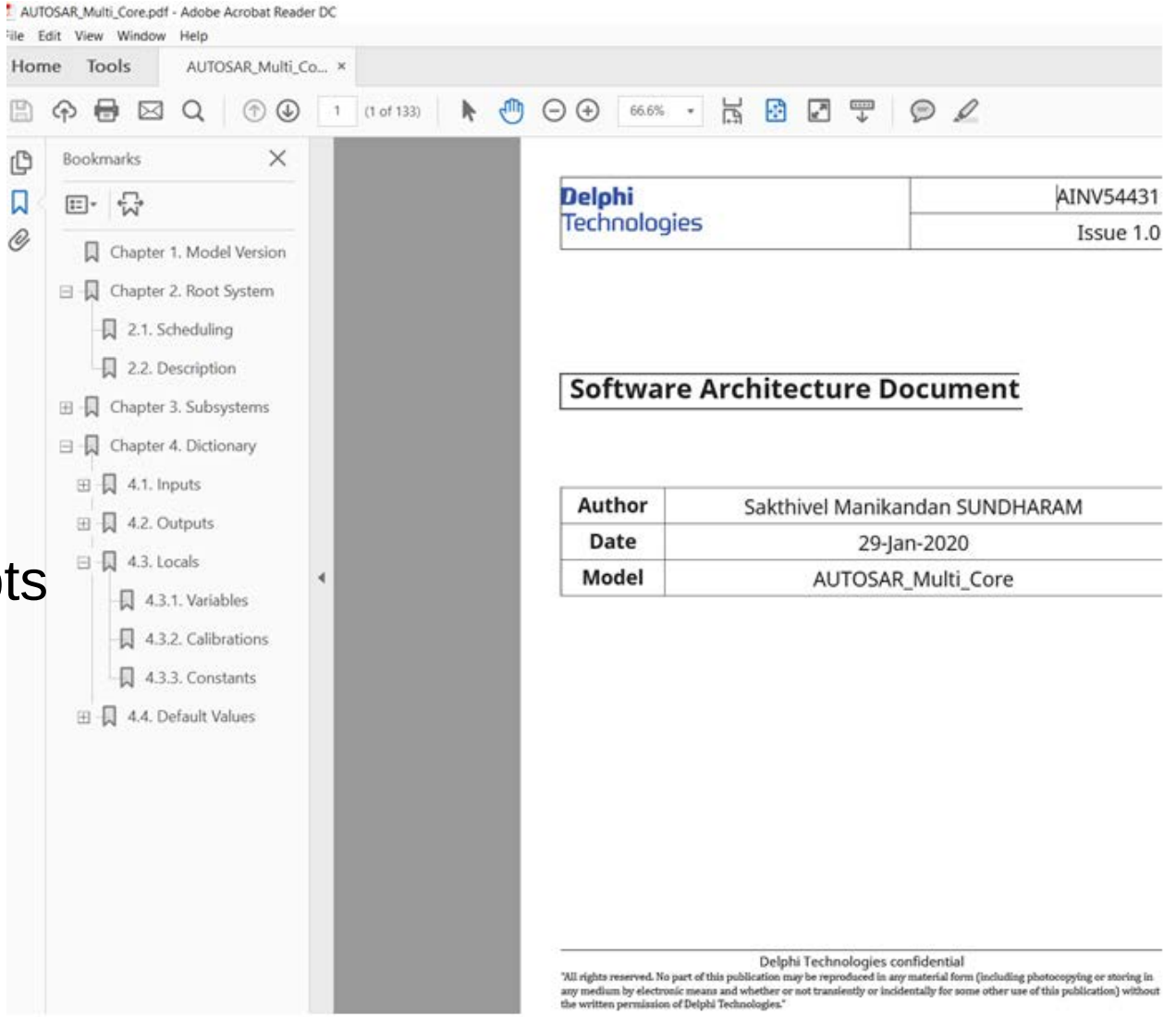


Delphi Technologies

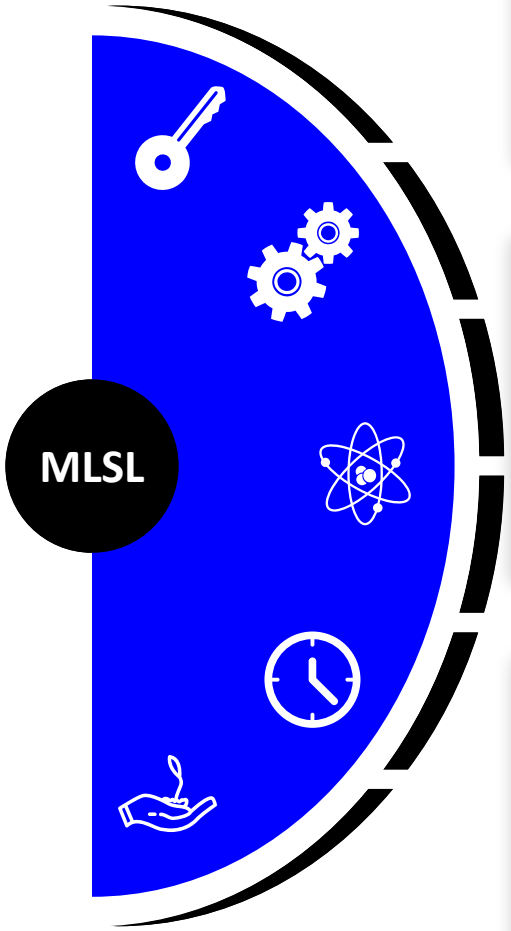
Generation of SW Architecture Documents



Automated Scripts



Lessons Learnt and Best Practices



AUTOSAR SW Architecture Authoring

- Modeling of AUTOSAR-based system architecture using AUTOSAR blockset together with System composer toolbox in recent releases of Matlab/Simulink.
- Creating fine-grained AUTOSAR architecture models using Simulink System Composer data dictionary support.

Requirements to SW architecture mapping

- Employing seamless approach to establish bidirectional traceability between modeling environment and the requirements database. Tracking of requirements back and forth between both the environments to verify fulfillment of requirements.
- To publish requirements and design on to requirements database. Also, the approach updates both requirements and design whenever adapted for changes due to technical discussions in a more efficient way.

Architectural simulation and SAD

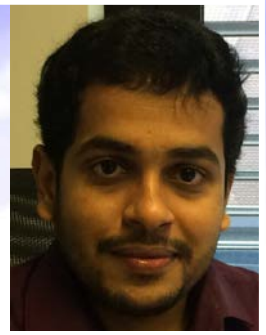
- Import and export of ARXMLs between architectural modeling environment to Basic software (BSW) configuration and development tool-chain to reduce ambiguity on architectural considerations and development time.
- Early model-based performance and trade-off analysis of non-functional requirements using custom-defined profiles (e.g. employing Matlab/Simulink and System Composer toolboxs).

30 June 2020

AUTOSAR Architecture Modeling of Multi-core Electric Powertrain Controller

Dr Sakthivel Manikandan Sundharam /
Software Architect

Delphi Technologies



Q & A