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
Delphi Technologies Inverter – The 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

### Gravimetric power density (kVA/Kg)
- **Conventional**
  - Volume (L): 10.7
  - Mass (Kg): 14.5
- **Delphi Technologies Inverter with viper**
  - Volume (L): 7.6
  - Mass (Kg): 8.4

### Volume (L)
- **Conventional**
  - 10.7
- **Delphi Technologies Inverter with viper**
  - 7.6

### Mass (Kg)
- **Conventional**
  - 14.5
- **Delphi Technologies Inverter with viper**
  - 8.4

### Efficiency Improvement (MPG)
- **Conventional**
  - 0
- **Delphi Inverter next generation**
  - 2

- **Many, many wire-bonds**

- **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

Multicore Electric Powertrain Controller
Pitfalls in Legacy Approach and Best Practices Evaluated

1. **Dearth of tool support for fine-grained modeling of AUTOSAR-based architecture design**
2. **Lack of support for proof-of-concept study of what-if scenarios, performance analysis**
3. **Gaps in requirements traceability**

1. **AUTOSAR and Simulink System Composer toolboxes for AUTOSAR-based systems engineering**
2. **Proof-of-concept studies, performance and viewpoint analysis of various stakeholders**
3. **Bidirectional traceability**
Static Software Architecture Tooling Twins

Legacy SysML

MLSL’s AUTOSAR Blockset + System Composer

Interface definition and Data Dictionary
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

Tracking of requirements back and forth between modeling and requirements database to verify fulfillment of requirements.
Bi-directional Traceability - Forward

Requirements to Architecture Traceability
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.
arxml Import from BSW Tools (f.e Vector BSW-stack Tools)
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 toolboxes).