E-Mobility System and Software Engineering

Vector Live Webinar
Agenda

- **E-Mobility System Development Challenges**
  - Model based System Design
  - E-Mobility Architecture Development with Traceability and Consistency
  - Toolchain for Software Design
  - System Requirement Generation
E-Mobility System Development Challenges

From Traditional to a Library-based Development Approach

**Traditional development approach**
- Development per project
- Low reuse = High variance
- High effort in project

**Library-based development approach**
- Define & manage reusable elements in global library
- Use of global library in projects = reuse & standardization
- Reduced effort in project

---

Transition

![Project A](image1)

![Project B](image2)

![Project C](image3)
E-Mobility System Development Challenges

From Document-based to Model-based Development

**Document-based development**
- Developing in numerous documents
- High effort in keeping documents in sync
- Difficult traceability and consistency

**Model-based development**
- Design systems once → reuse in end-to-end design
- Interconnected model from REQ-System-SW-HW-COM
- Increased efficiency, traceability and consistency

Transition
Agenda

E-Mobility System Development Challenges

- **Model based System Design**
  
  E-Mobility Architecture Development with Traceability and Consistency
  
  Toolchain for Software Design
  
  System Requirement Generation
Model based System Design

PREEvision Modeling Overview
Model based System Design

Product Line Engineering Concept

- Platform Library manages E-Mobility system Components and Interfaces
- System Library manages selectable system variants for application in specific vehicle projects.
- Vehicle Product Line consists of specific system variant with its sensors, applications and actuators for a specific vehicle.

PREEvision Database

Platform Library
- Full E-Mobility Planform Model (150%)
  - Sensing / Actuation Component
  - Specific Application/Arbitration
  - Blocks/Data types/Interfaces

System Product Line
- Partial E-Mobility System Model (120%)
  - Variants of Specific System
  - Variants of Specific Vehicle
  - Partial Communication Artifact

Vehicle Product Line
- Specific Vehicle System Model (100%)
  - Specific Component
  - Specific Vehicle System
  - Specific Network communication

* PL: Product Line
* M: Mapping
Model based System Design

SW Platform Reuse Concept

E-Mobility Platform

Software Component Types
- APS BPS Selection
- Driver Status Estimation
- Processing Data
- BAT Power Calculation
- BMS Relay Operation

Software Composition Type
- Driver Intention Analysis
- Fuel Cell Monitoring
- Power Distribution
- HV Battery Mgmt.
- HEV Common Function

Software Composition Prototype
- Driver Intention Analysis
- APS BPS Selection
- Driver Status Estimation
- Fuel Cell Monitoring
- Stack Voltage Calculation
- Processing Data
- Power Distribution
- BMS Relay Operation
- Main Relay OFF Cnd
- HV Battery Management
- BAT Power Calculation
- Main Relay Actuation
- Motor TQ Generation
- Driver Status Estimation
- Motor TQ Calculation

System SW Product Line

HEV Architecture Design (120%)
- HEV Root Composition
- Driver Intention Analysis
- Power Distribution
- Motor TQ Generation

“EB1” Vehicle Product Line

HEV’ SW Architecture Design (100%)
- HEV Root Composition
- Power Distribution
- Motor TQ Generation

“NX8” Vehicle Product Line

FCEV SW Architecture Design (120%)
- FCEV Root Composition
- Driver Intention Analysis
- Power Distribution
- Fuel Cell Monitoring
- Motor TQ Generation
- HV Battery Mgmt.

FCEV’ SW Architecture Design (100%)
- FCEV Root Composition
- Power Distribution
- Motor TQ Generation

Reuse and Derive Variants
## Mappings

### Customer Feature / Requirements
- **E-Mobility Feature**
  - Electric Power Source
  - High Voltage Battery
  - Small Capacity Battery
- **E-Mobility Requirement**
  - H2 Supply
  - FC Monitoring
  - FC Voltage Check

### Logical Architecture
- **Input**
  - SNSR
- **Logic**
  - FUNC
- **Output**
  - ACT

### SW Architecture
- **Platform SWC**
  - BAT
  - PRS
- **Application SWC**
  - COV
- **Internal Behavior**
  - Runnable

### HW / NET Architecture
- **Sensor**
  - SNSR
- **ECU**
  - MCU
- **Actuator**
  - MTR

### Communication
- **Signal**
- **PDU**
- **Frame**
- **Gateway**

### Model based System Design

#### Customer Feature / Requirements Design
- Feature and variant definition for all components of E-Mobility systems
- Requirement definition for each function

#### Logical Architecture Design
- Describes functionality of Customer Features and Requirements for implementing software and abstracts them in the form of input-function-output to clarify

#### Software Architecture Design
- SWC creation and Interface (including data type) connection for function implementation
- Definition of hierarchical structure and internal behavior of SWC

#### Hardware / Network Architecture Design
- Define all HWC of the system
- Create connections to define the Network Topology
- Distribute functionality by mapping SWC to HWC

#### Communication Design
- Create signals based on SWC port information, allocated to HWC in Network Topology
- Create transmissions by synthesizing PDU and Frame
Model based System Design

Derivate product line using variant management

- Create Customer Feature relations using Variant Diagram
- Derive a new product line to create a vehicle-specific system using the propagated artifacts
- Propagate artifacts that are mapped to the Customer Features into a Architecture Variant container using Propagation Rule
- Configure Alternatives by selecting Customer Features using variant templates
Agenda

E-Mobility System Development Challenges
Model based System Design

- **E-Mobility Architecture Development with Traceability and Consistency**
  Toolchain for Software Design
  System Requirement Generation
E-Mobility Architecture Development with Traceability and Consistency

Customer Feature and Requirement Design

- E-Mobility Platform Feature List is created by separating common and variation point of each system
- Through Feature Condition Diagram, user can select effective feature combination and reduce human Error
- System Requirement List describes the technical requirements or the regulations of the system
Logical Architecture Design

- Logical Architecture describes the abstracted e-mobility functions of the entire system.
- End-to-end design including the system to be developed and the functions of the cooperative control system.
- Before technical implementation, define the system scope and role by using abstract functions and interfaces.

**Motor Torque Generation Function**

**H2 Tank Management Function**

**H2 Supplement Function**
Software Architecture Design

- Software Architecture defines system functions and interfaces at the technical implementation level.
- Software components are instantiated and reused based on the types of E-platform library.
- Each software component or composition is mapped to the ECU to be implemented.
In hardware architecture, communication and hardwire connections between sensors, ECUs, and actuators are described.

Network topology describes high voltage power distribution lines, converters and batteries of E-Mobility system.

The connection between cooperative ECUs are describes on the topology to enable hardware End-to-End design.
E-Mobility Architecture Development with Traceability and Consistency

Communication Design

AUTOSAR or DBC Import
SW Design
HW Design
SW/HW Mapping
Data Mapping
Signal Routing
Communication Design
Layout, Routing,
Bus Access
AUTOSAR Export

Signal Assignment to Frame (Message)

Signal Routing
Path Creation
Customer feature selection enables deployment from E-Mobility Platform to each EV, HEV, and FCEV System Product Line.

Corresponding artifacts mapped to the selected feature can be activated or deactivated through variant activation.

This can be applied not only to the system but also to the entire vehicle, and the engineer in charge of each vehicle system can continue the detailed design within the deployed product line.
E-Mobility Architecture Development with Traceability and Consistency

PREEvision Tool Demo
Agenda

- E-Mobility System Development Challenges
- Model based System Design
- E-Mobility Architecture Development with Traceability and Consistency

- **Toolchain for Software Design**
  - System Requirement Generation
Toolchain for Software Design

Toolchain for SW Design

System Design / PREEvision
- Top-Down Approach
  - Architecture Design
  - SWC Internal Behavior Design

Export
- System Desc.

Import
- Round-Trip
- ASW Generation

Implementation / e.g. Matlab
- System Model Design
- Algorithm Design

ASW Code(.c/.h)

Configuration / e.g. DaVinci
- Bottom-Up Approach
  - ECU Extract

Update
- SWC Desc.

Compile & Test / e.g. Canoe
- RTE / BSW Generation
- Compile & Debug
- Target Testing

BSW Code(.c/.h)
Video Tutorial

PREEvision & Simulink Toolchain Tutorial Video
Agenda

- E-Mobility System Development Challenges
- Model based System Design
- E-Mobility Architecture Development with Traceability and Consistency
- Toolchain for Software Design
  - System Requirement Generation
Automatic Report Generation

- Report creation using company's standard template
- Automatic configuration of diagram or table layout using mapping relation
PREEvision Report
Generation Tutorial Video
PREEvision – This is Model-Based E/E Engineering

www.vector.com/preevision

Author:
Kim, Eui-yeul
Vector Korea