Variability Modeling in the Automotive Domain
Past, Present and Future Approaches

Stefan Kuntz
Introduction
The Key Note Speaker

Group Leader SW Architecture

Stefan Kuntz
Powertrain - Engine Systems
PES GE HSA SWRBG

Phone: +49 (941) 790-70073
E-Mail: Stefan.Kuntz@continental-corporation.com
Agenda

1. Automotive Domain
2. Past
3. Present
4. Future
5. Conclusion
Automotive Domain
Significant Driving Forces

› Cost sensitive
› Resource constraints
› Short time-to-market

› Increasing functionality
› Larger scope of distribution of functionality
› Tighter collaboration among organizations
› High degree of division of labor
Continental Corporation
Five Strong Divisions

Chassis & Safety
- Vehicle Dynamics
- Hydraulic Brake Systems
- Passive Safety & Sensorics
- Advanced Driver Assistance Systems (ADAS)

Powertrain
- Engine Systems
- Transmission
- Hybrid Electric Vehicle
- Sensors & Actuators
- Fuel & Exhaust Management

Interior
- Instrumentation & Driver HMI
- Infotainment & Connectivity
- Intelligent Transportation Systems
- Body & Security
- Commercial Vehicles & Aftermarket

Tires
- PLT, Original Equipment
- PLT, Repl. Business, EMEA
- PLT, Repl. Business, Asia Pacific
- Commercial Vehicle Tires
- Two Wheel Tires

ContiTech
- Air Spring Systems
- Benecke-Kaliko Group
- Compounding Technology
- Conveyor Belt Group
- Elastomer Coatings
- Fluid Technology
- Power Transmission Group
- Vibration Control

Automotive
PLT – Passenger and Light Truck Tires
Automotive Domain
Powertrain Engine Systems: Diesel DI
Automotive Domain
Example: Combustion Engine Management System

› Roughly over 6,000 system/engine variants
› Software:
  › 70 up to 140 functionalities
  › 4,000 to over 8,000 executable units
  › 20,000 to over 42,000 data/variables
  › 1.5 to 2.2 MByte of ROM
  › 0.75 to 1.5 MByte of RAM
› Calibration data

Hardware/ECU:
› Minimum 7,000 variants
Automotive Domain
Example: Smart NOX Sensor

› Roughly 200 Variants
  › Three different micro controller families
  › Different standard and proprietary communication protocols
  › Various functionality including third-party functionalities
  › Diagnostics

› Number of NOX sensors in a vehicle - NOX Sensor Network

› Passenger and commercial vehicles
Variations shall be managed

› at different levels of abstraction, and

› at various activities in the development and maintenance process

It is an architectural topic: Requirement - Structure - Variability
Agenda

1. Automotive Domain
2. Past
3. Present
4. Future
5. Conclusion
Past
Specification of Variation

› Manual approach
› Document based supported by proprietary tool chain - documents and tables
› Verification and validation by reviews - Four-Eyes Principle
› Detection of inconsistencies at build/compile time respectively specific testing/inspections

Specification of variation - Definition of Variants

1.4 Version and Configuration

<table>
<thead>
<tr>
<th>Versions:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Name</td>
</tr>
<tr>
<td>------------</td>
</tr>
<tr>
<td>NC_ENSD_VERS</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Legend:
HED Hybrid Electric Drives
NC Non Configurable

Programming Language Artifact

```c
int function ( ... ) {
    ...
    <statements> ...
    #if NC_ENSD_VERS == 1
    ...
    <statements> ...
    #endif
    ...
    <statements> ...
    return( OK );
}
```
Agenda

1. Automotive Domain
2. Past
3. Present
4. Future
5. Conclusion
Present
Levels of Abstraction
**Present**

**EAST-ADL - Purpose of Abstraction Levels**

<table>
<thead>
<tr>
<th>Level of Abstraction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Vehicle Level</strong></td>
<td>This level describes the features visible to the stakeholder “driver” such as windscreen wipers, window lifter, cruise control, etc, as well as the dependencies between these features.</td>
</tr>
<tr>
<td><strong>Analysis Level</strong></td>
<td>This level captures the external visible behavior and algorithms of the functionality, as well as the inter-dependencies between these functionalities. “What the system shall do?”</td>
</tr>
<tr>
<td><strong>Design Level</strong></td>
<td>This level represents the realization of each functionality analyzed on the Analysis Level. It represents the logical architecture. “How the system is doing what it shall do?”</td>
</tr>
<tr>
<td><strong>Implementation Level</strong></td>
<td>This level describes the implementation of the functionality described on the Analysis and Design Level. It represents the technical architecture, and consists of software, hardware and mechanics.</td>
</tr>
</tbody>
</table>
Present

EAST-ADL - Cross-Cutting Concerns

Vehicle Level

Analysis Level

Design Level

Implementation Level

Level of abstraction

Cross Cutting Concern

Requirements

Variability

Dependability/Safety

Timing

...
Present

EAST-ADL: Feature Modeling

Vehicle Level

Technical Feature Model

Analysis Level

Functional Analysis Architecture

Environment Model

Feature Model

Design Level

Functional Design Architecture

Feature Model

Implementation Level

AUTOSAR VFB, Software Component, System, Basic Software Module, and ECU view

Allocation

Hardware Design Architecture

Feature Model

AUTOSAR ECU Resource Description

Level of abstraction

Models

Transformation
Present

EAST-ADL: Feature Modeling Meta Model

Variability modeling in the automotive domain
Public

2015-01-21
Stefan Kuntz, © Continental AG
Present

Brief Introduction into AUTOSAR - Objectives

› Standardization of basic software functionality of automotive ECUs
› Scalability to different vehicle and platform variants
› Transferability of software
› Support of different functional domains
› Definition of an open architecture
› Collaboration between various partners
› Development of highly dependable systems
› Support of applicable automotive international standards and state-of-the-art technologies
Present

Brief Introduction into AUTOSAR - Methodology

<table>
<thead>
<tr>
<th>System - OEM</th>
<th>Electronic Control Unit (ECU) - Supplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Views:</td>
<td>Views:</td>
</tr>
<tr>
<td>• Virtual Function Bus (VFB)</td>
<td>• ECU</td>
</tr>
<tr>
<td>• Software Component (SW-C)</td>
<td>• Basic Software (BSW) Module</td>
</tr>
<tr>
<td>• System Topology</td>
<td></td>
</tr>
</tbody>
</table>

System Configuration Generator

ECU Extract of System Configuration

ECU Configuration Generator

Runtime Environment (RTE)

Operating System (OS)

Communications (COM)

BSW Modules (BSWM)

µC Abstraction Layer (MCAL)

Software Components (SW-C)
Present

AUTOSAR Support for Variability

› Variation Point
  › AUTOSAR meta model: stereotype «atpVariation»
  › Controlled by System Constants
  › Aggregation
  › Association
  › Attribute Value
  › Property Set

› Feature Model
  › Specific AUTOSAR template
  › Feature Selection and Feature Map
Present

AUTOSAR Variation Point

VariationPoint

+ shortLabel :Identifier [0..1]

+ swSyscond [0..1]
  SwSystemconstDependentFormula
    ConditionByFormula
    + bindingTime :BindingTimeEnum

+ blueprintCondition [0..1]
  DocumentationBlock

+ formalBlueprintCondition [0..1]
  SwSystemconstDependentFormula
    «atpMixedString»
    BlueprintFormula

+ postBuildVariantCondition [0..*]

+ matchingCriterion [1]
  ARElement
    AtpDefinition
    PostBuildVariantCriterion

+ relatedBindingTime = preCompileTime

+ elaboration
  elaborationCondition

+ relatedTime
  relatedTimeValue

+ enum
  «enumeration»
  BindingTimeEnum
  SystemDesignTime
  codeGenerationTime
  preCompileTime
  linkTime

+ relatedEnum
  «enumeration»
  AdditionalBindingTimeEnum
  blueprintDerivationTime
  postBuild

Courtesy AUTOSAR Development Cooperation 2015
AUTOSAR R 4.2.1
Present

AUTOSAR Feature Model

› Feature Model - Hierarchical model containing features and its dependencies
› Feature Selection - Decision model
› Feature Map - Mapping features and system constants

Feature Model: Sample Car

Mandatory: Engine
  ▶ Alternative: Gasoline Engine
  ▶ Alternative: Diesel Engine
  ▶ Requires: Gasoline Engine Controller
  ▶ Requires: Diesel Engine Controller

Mandatory: Engine Controller
  ▶ Alternative: Gasoline Engine Controller
  ▶ Alternative: Diesel Engine Controller

Mandatory: Doors
  ▶ Mandatory: Two Doors
  ▶ Optional: Four Doors

Optional: Extras
  ▶ Optional: Convertible
  ▶ Optional: Sunroof
  ▶ Mandatory: Electric window lift
  ▶ Optional: Halogen lights

Conflicts: Sunroof
Conflicts: Four Doors

Courtesy AUTOSAR Development Cooperation 2015
AUTOSAR R 4.2.1
Present

AUTOSAR Feature Model - Meta Model

Courtesy AUTOSAR Development Cooperation 2015
AUTOSAR R 4.2.1
Present
AUTOSAR Example

Sensor SW-C
SW-C #2

Application SW-C
SW-C #3

Application SW-C
SW-C #8

Application SW-C
SW-C #4

Actuator SW-C
SW-C #5

ECU HW Abstraction Sensor
SW-C #1

AUTOSAR Service
SW-C #7

ECU HW Abstraction Actuator
SW-C #6

Variability modeling in the automotive domain
Public

2015-01-21
Stefan Kuntz, © Continental AG
Present

AUTOSAR Example … continued

Variability modeling in the automotive domain

2015-01-21
Stefan Kuntz, © Continental AG
Present
AUTOSAR Example … continued
The automotive industry is making good progress and works towards a standard
<table>
<thead>
<tr>
<th></th>
<th>Agenda</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Automotive Domain</td>
</tr>
<tr>
<td>2</td>
<td>Past</td>
</tr>
<tr>
<td>3</td>
<td>Present</td>
</tr>
<tr>
<td>4</td>
<td>Future</td>
</tr>
<tr>
<td>5</td>
<td>Conclusion</td>
</tr>
</tbody>
</table>
Future Challenges

› Heterogeneous solutions for methods and tools

› There is still no seamless integrated tool chain supporting variations across all levels of abstraction

› Variability is still not considered as integral part of architecture: Requirements - Structure - Variability

› Lack of tools to migrate existing assets: “Harvesting Variability”

› Lack of tools for analysis of variations and variability (visualization)

› Determination of reusable packages
Future
Seamless Integrated Tool Chain

System

Function

Software

Hardware

Mechatronics

Feature Models

Feature Models

Feature Models

Variability modeling in the automotive domain
Public

Continental

Stefan Kuntz, © Continental AG

2015-01-21
Future

Idea: Configuration Management System and Variability

› Checking out version and variant and vice versa
Future
Idea: Integration Tool
Future
Idea: Integration Tool … continued
Future

Idea: Integration Tool … continued
Conclusion

- Theory of variations respectively variability is already understood in the industry
- Lack of integrated and seamless tool chain, for
  - Defining and specifying variations
  - Analyze variations
  - Process variations
  - Visualize variations depending on context
Thank you very much for your attention!