Some Challenges for Automotive Embedded Systems

Matthias Weber
Carmeq GmbH
Overview

Carmeq

Model-Based Development
Requirements Specification
Product-Lines / Reuse of Development Artifacts
Mission

Our mission is technical consulting and engineering services focused on software-driven systems for the automotive industry.

We improve quality and reduce costs through customer-oriented use of advanced technologies, efficient development processes and modern architecture.
Carmeq - Past and Present

04 June 2002
  Decision to found Carmeq by the group's board of directors

30 July 2002
  Carmeq GmbH founded as a 100% subsidiary of the Volkswagen Group

01 January 2003
  Business commences with 16 employees

Today (September 2008)
  Approx. 220 employees

Sites
  Berlin (head office)
  Wolfsburg
  Ingolstadt
Basics of Model-Based Development
Traditional Approach

Traditional development process based on documents

- Textual specification of functions
- Manual Implementation of (simulation) prototypes or production code
- Late Testing
Model-Based Approach

Model-based development process

- (almost) continuous presence of executable functional models
- (almost) continuous validation and testing
- Possibility of automatic compilation into C-Code
The Challenge

Product Related Challenges
- Functionality increase
- Complexity increase
- Increased Safety-criticality
- Quality concerns

Challenges Related to Development Process
- Supplier-OEM relationship
- Multiple sites & departments
- Product families
- Componentization
- Separation of application from infrastructure
- Safety Requirements, ISO 26262
Which Models should be used?

- The use of modeling languages or notations has become standard practice in almost all engineering disciplines.

- In the automotive domain, electronics (control systems) and computer science (software) have grown to dominating importance.

- There is a desire to use a single modeling language in order to avoid semantic ruptures or even inconsistencies.

- Preconditions:
  - The modeling language is sufficiently powerful to model all relevant aspects and to provide adequate views
  - The modeling language is understood by all stakeholders, at least in those parts relevant for the respective stakeholder
  - There are appropriate methods and tools available for modeling (and simulation)
### Examples of Modeling Languages

<table>
<thead>
<tr>
<th>Sprache</th>
<th>Beispiel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block Diagrams</td>
<td><img src="image" alt="Block Diagram" /></td>
</tr>
<tr>
<td>State automata (including Harel's extensions)</td>
<td><img src="image" alt="State Automata Diagram" /></td>
</tr>
<tr>
<td>UML/SYSML</td>
<td><img src="image" alt="UML Diagram" /></td>
</tr>
<tr>
<td>Domain specific Architectural languages&lt;br&gt;EAST ADL, Autosar</td>
<td><img src="image" alt="Domain Specific Diagram" /></td>
</tr>
</tbody>
</table>
The Response - EAST-ADL2

Architecture Description Language for Handling all engineering information required to sustain the evolution of vehicle electronics
EAST-ADL2

A System Modeling Approach that
Is a template for how engineering information is organized and represented
Provides separation of concerns
Embrace the de-facto representation of automotive software – AUTOSAR

Feature content
Abstract functional architecture
Functional architecture, HW architecture, platform abstractions
AUTOSAR Software architecture
Embedded system in produced vehicle
EAST-ADL2 – AUTOSAR Mapping

Design Level

Implementation Level

Realizes
Timing Measures

Input
Synchronization: What is the difference in time between a number of inputs that eventually joins

Example ABS:
What is the maximum difference in time between the “concurrent” samples of the four wheel sensors.
Necessity of Modeling Guidelines

Modeling Guidelines for MATLAB/Simulink/Stateflow

Catalog of Rules
After invocation (power-on), the interior light shall be off.

- Opening one or both doors invokes the light, which dims up within 1 second in 10 steps.
- If both doors are closed, the light shall dim to off (1 second, 10 steps).
- If the light is on for 5 minutes without any driver action (i.e. opening or closing a door), the light shall dim down (for power-saving reasons).
Requirements Specification
Requirements Specification: OEM-Supplier Contract

Responsibility: OEM

SW Development: Supplier or OEM

Acceptance Test
Responsibility: OEM

Requirements Specification

OEM-Supplier Contract
Model-Based Development

Textual Requirements are indispensable

- Executable models focus on constructive aspects, i.e. important information cannot be modeled adequately
  - High-level Requirements
  - Non-functional requirements,
  - System properties
  - Rationale for requirements
  ...

- Further documentation is indispensable
  - However: system requirements $\neq$ model documentation

- Requirements from standards (e.g. SPICE):
  - Separate requirements phase
  - Requirements tracing across all development phases
EE Specification Volume - Mercedes S-Class (W220)

500 Distributed Winword Documents
Typical Questions in a Project Context

- Where is the latest version of requirement X.
- Have the requirements for function X been reviewed by the supplier?
- Which requirements are implemented by ECU X.
- Which ECU-sample should realize which requirements? Have the suppliers agreed to it?
- What has been changed for function X since the last review? Who did these changes?
  - What kind of impact do these changes have on the tests?
  - What are the costs for these changes?
- Which requirements have been deleted? Which have been postponed until later versions?
Requirements Management: Documents versus Data Base

Paper (Winword) documents: up to 500 per production series

Requirements specification = total of all documents

RM tool data base: 1 per production series

Requirements specification = total of all requirements

RM tool manages text modules as individual requirements (objects)

Documents are created as extracts from the database
Exchange of Specifications

Hersteller-modul

Austausch-modul

Zulieferer-modul

H6

change due to up-sync

H7

edit

...
There is no clear boundary between manufacturer requirements specification and supplier system specification!

Customer also has to specify significant parts of the solution.

Customer demands and contractors duties in automotive development.
Relation between Model and Requirements – Ideal World

**Requirements**

<table>
<thead>
<tr>
<th>Funktionsbereich (z.B. Licht und Beleuchtung)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Funktionsfamilie (z.B. Blinken)</td>
</tr>
<tr>
<td>1.1 Bauteilfunktion (z.B. Richtungsblinker)</td>
</tr>
<tr>
<td>…</td>
</tr>
<tr>
<td>1.2 Bauteilfunktion (z.B. Warnblinken)</td>
</tr>
<tr>
<td>…</td>
</tr>
<tr>
<td>2. Funktionsfamilie (z.B. FahrBremsStandlicht)</td>
</tr>
<tr>
<td>1.1 Bauteilfunktion</td>
</tr>
<tr>
<td>…</td>
</tr>
</tbody>
</table>

**model**

- Blinken
- FahrBremsStandlicht
- Light System

<<umgesetzt durch>>
Relation between Model and Requirements – Real World

Light System

Model (for series code generation)

Requirements

Funktionsbereich (z.B. Licht und Beleuchtung)

1. Funktionsfamilie (z.B. Blinken)
   1.1 Bauteilfunktion (z.B. Richtungsblk)
   ... 
   1.2 Bauteilfunktion (z.B. Warnblinken)
   ... 

2. Funktionsfamilie (z.B. FahrBremsStandlicht)
   1.1 Bauteilfunktion
   ...
Product Lines / Reuse of Development Artifacts
Market Segmentation

- Number of segments is increasing; size is decreasing.
- The significance of individual models is decreasing – product families are of growing importance.

“Leohold: Herausforderung zwischen Funktionsintegration und Komplexitätsmanagement”
Development Artifacts of an Automotive Electronic System

- >100k requirements
- >100k funktion blocks
- >100k test cases
- hundreds of other artifacts

- objects

- >100k requirements

- >100k funktion blocks

- >100k test cases

- hundreds of other artifacts
Product lines for Specifications / Models / Tests / Code etc. „Real World“

- Legacy artifacts
- Domain knowledge
- Innovations

Feedback / Carry-over of specific solutions (bottom-up reuse)

Domain Engineering (project-independent)

Reference artifacts (top-down reuse)

Specifications, Models, Test cases, SW-Components, …

Referencing Artifacts

Product Configuration

New requirements

Product Development (project dependent)
Reuse in the automotive domain

- High Degree of Variability
  - Car Platforms
  - Markets
  - Variant and Optional Functionality
  - Different Laws and Regulations (geographical, temporal)
  - Different Availability of Parts (geographical, temporal)
  - Technology changes
  - Cost pressure

- Heterogeneity of run-time environment (Hardware & Software)

- Long product lifecycles (ca. 5+20 Jahre)

- Diverging lifecycles (e.g. infotainment vs. safety-relevant functions)

⇒ Reuse is very difficult but indispensable
Herausforderungen für Volkswagen.
Beispiel Freisprecheinrichtung.

- Different ECU Variants
- Different HMI concepts
- Different Cell-Phones
- Different Cell-Phone Adapters
- Different Software-Platforms
- Short Development Cycles in the Cell-Phone market with sharp rising functionality
### Problem Areas.

<table>
<thead>
<tr>
<th>Module Strategy</th>
<th>Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of precision of module concept wrt. module variability and configuration</td>
<td>Ad-hoc reuse of development artifacts leads to massive increase in variants</td>
</tr>
<tr>
<td>No clear identification of causes of variability</td>
<td>Rising variability after SOP base on technical changes and new parts</td>
</tr>
<tr>
<td>Insufficient basis of variability-related decisions</td>
<td>insufficient backward compatibility</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Production</th>
<th>After-Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration complexity in production keeps growing ⇒ existing production logistics becomes insufficient.</td>
<td>Variant-Complexity leads to increased storage costs to meet after-sales requirements</td>
</tr>
<tr>
<td>Specifics of production, such as use of left-over parts lead to many new variants in the field.</td>
<td>Rising number of variations in car-fleet based on flashware and parameterisations</td>
</tr>
</tbody>
</table>

**Challenges**

**Variant-Management in Automotive Electronics**
Abstimmung Variantenmanagement@EE.

Bereichsübergreifendes Variantenmanagement

Vehicle
- Module A
- Module B
- Module C
- Alternative 1
- Alternative 2
- Option

Bewertung/Freigabe übergreifender Varianten

Bereichspezifisches Variantenmanagement

Produktion
- Entwicklung
- Produktmanagement
- After Sales

Bewertung/Freigabe bereichsspezifischer Varianten

Beiträge des Zulieferers
Integration of parallel Innovations / Introduction of product line development

Common Artefacts of Products A and B

Extraction

Motivation: The integration of parallel innovations should be supported!

Development of System A

Development of System B
Further Challenges

- Timing behaviour
- Error modeling
Thank You

Matthias Weber
Tel. +49-30-3983 537 230
e-mail: matthias.weber@carmeq.com
Backup - EAST-ADL2
Outline
• Example usage of EAST-ADL2
• Model Structure
• Example Model
• AUTOSAR Relation
• Areas covered by EAST-ADL2
• Conclusion
Some Typical Scenarios

The Vehicle Manufacturer decides what to include in the next product
A Chassis engineer analyses a novel control algorithm
Application expert defines detailed design
Software engineer defines software architecture
Packaging and allocation, Integration on ECU
Early phase validation and verification
Product Planners decide what to put in the next product

Features represent the properties/functionality/traits *(Brake, Wiper, CollisionWarning, …)*

Vehicle Feature Model organize Features for the vehicle

Variability mechanism supports the definition of rules for inclusion in different vehicles – Product Line Architecture
A Chassis engineer analyses a novel control algorithm

Control algorithm is defined as a ADLFunction connected to a plant ADLFunction in the Environment model

EAST-ADL2 defines structure, legacy tools can be used for behavior definition, simulation, etc.

Realization details are omitted:
Functional validation and verification can be done with respect to key aspects
Understanding of key aspects is possible
An OEM and Supplier agree on specification

A model of the supplied system provides a clear and effective information exchange
Functions can be integrated and validated before SW and HW exists
Interfaces and interaction is clear, avoiding common specification bugs
Application expert defines detailed design

A detailed functional architecture is defined, addressing e.g.

- Hardware architecture
- Allocation
- Fault tolerance
- Implementation concerns
- Sensor, actuator constraints

Focus is behavior and interaction of functions
Software engineer defines SW Architecture

AUTOSAR Application SW Components are defined
The set of SW components together realizes the Functional Architecture
Software organization and functional organization is decoupled and optimization of the SW architecture is possible.
Legacy, sourcing, allocation, performance, verification, responsibility, re-use, etc. influence which functions are realized by each SW component
Outline

• Example usage of EAST-ADL2
• Model Structure
  • Example Model
  • AUTOSAR Relation
• Areas covered by EAST-ADL2
• Conclusion
EAST-ADL2 System Model

System Model

- Vehicle Feature Model
- Analysis Architecture
  - Functional Analysis Architecture
- Design Architecture
  - Functional Design Architecture
  - Middleware Abstraction
  - Hardware Design Architecture
- Implementation Architecture
  - AUTOSAR System
- Operational Architecture

Vehicle Level
Analysis Level
Design Level
Implementation Level
Operational Level
Principle of Realization

- Entities on lower abstraction level realizes Entities on higher abstraction level
Outline

• Example usage of EAST-ADL2
• Model Structure
• **Example Model**
• AUTOSAR Relation
• Areas covered by EAST-ADL2
• Conclusion
Function interactions – end-to-end

- Model structure supports interaction with the environment and end-to-end functional definitions
Hardware Design Architecture

- Hardware architecture to allow hardware design and functional allocation
- Behavior of HW entities can be defined for analysis of end-to-end function
Outline

• Example usage of EAST-ADL2
• Model Structure
• Example Model
• AUTOSAR Relation
• Areas covered by EAST-ADL2
• Conclusion
EAST-ADL2 Complements AUTOSAR

- EAST-ADL2 is an information structure including aspects beyond the Software Architecture
  - Requirements, traceability, feature content, variability, safety, etc.
- **Provides means to define what the software does**
  - An AUTOSAR specification defines the software architecture and information required for SW integration - but is neutral to its functionality
- **Provides means to model strategic properties**
  - Key vehicle aspects is captured independently of the software architecture
- **Supports modelling of error behavior and the representation of safety-related information and requirements**
EAST-ADL2 – AUTOSAR Mapping

Design Level

<<ADLFunction>> C2

ADLFunction E4

ADLFunction E5

<<ADLFunction>> C1

ADLFunction E1

ADLFunction E2

<<ADLFunction>>

C2

Implementation Level

Runnable E1

Runnable E4

Runnable E3

Runnable E2

ApplicationSWC A1

ApplicationSWC A2

Environment

Vehicle Level
Analysis Level
Design Level
Implementation Level
Operational Level
EE Architecture

Realizes
Outline

• Example usage of EAST-ADL2
• Model Structure
• Example Model
• AUTOSAR Relation
• Areas covered by EAST-ADL2
• Conclusion
Variability

- Definition of Feature Content of Vehicle using Feature Trees
- Definition of Product Line in terms of mandatory and optional features for each vehicle category
- Definition of Variability rules for realization
- Optional/mandatory functions and components
- Definition on how to resolve variability based on feature content
Requirements and V&V

- Definition of Requirement modelling framework based on SysML
- Concepts for capturing requirements and components in same model
- Traceability between requirements, components and V&V
- V&V constructs to capture test case, test outcome, etc.
- Integration of RIF concepts (Requirement Interchange Format)
Error modelling & failure analysis

- Modelling Concepts for Hazards and Error Propagation
- Basis for Hazard Analysis and Fault Tree and Failure Modes and Effects Analysis
- Tool Interface for Automatic FTA/FMEA
Safety Aspects & ISO 26262

- ASIL Categorization through requirements
- Support for Safety Case – Use of model entities to argue safety
- Organization of information in line with ISO 26262
- Support for methods required by ISO 26262
Behavior

- Definition of Behavioral semantics to allow legacy tool integration
  - Ascet, Simulink, legacy code, etc.
- "Native” EAST-ADL2 definition of Behavioral semantics
- Definition of relation to AUTOSAR behavior
- Behavioral Semantics for Environment model (Plant)
Outline

• Example usage of EAST-ADL2
• Model Structure
• Example Model
• AUTOSAR Relation
• Areas covered by EAST-ADL2
• Conclusion
**Conclusion**

- EAST-ADL2 provides an information structure for design of automotive embedded systems
  - Architecture Description Language
- Use of abstraction levels is a fundamental concept
  - entities on lower levels *realize* entities on higher levels
- EAST-ADL2 is a fully aligned complement to AUTOSAR
  - AUTOSAR is the SW architecture definition enabling SW component integration on ECU
  - EAST-ADL2 supports the successful integration of AUTOSAR components
  - EAST-ADL2 supports additional engineering steps including *feature definition, requirements engineering, V&V, safety analysis, functional modeling/integration, product line engineering*