Webinar - AUTOSAR Tooling

Convenient AUTOSAR authoring using DaVinci and model-based design
Agenda

1. Vector AUTOSAR Tool Chain
2. Interaction with ModelingTools
3. Tool Versions and Features
4. Summary and Demo
Vector AUTOSAR Tool Chain

Tools Overview

Logical & Software Architecture Design
- PREEvision

Development of Application Software
- DaVinci Configurator Pro

Virtual Integration Platform

SWC Detailed Design
- DaVinci Developer

RTE Configuration
- BSW Configuration
- DaVinci Configurator Pro

Basic Software (BSW)

AUTOSAR ECU

RTE

Calibration
- CANape

ECU-/System-Test
- CANoe

ECU Monitoring and Debugging
- CANoe.AMD

Virtual Integration

Virtual Integration Platform
Vector AUTOSAR Tool Chain

DaVinci Developer

- Creation of SWC descriptions with graphical or table-based editors
- Definition of SWC internal behavior (runnable entities)
- Consistency check of the SWCs
Vector AUTOSAR Tool Chain

DaVinci Configurator Pro

- Creation of ECU configurations
- MICROSA BSW and RTE
- Third party BSW (MCALs)
- Specific editors for each BSW domain
- Configuring the RTE section
- Runnable to Task Mapping
- Validation of ECU configuration
- Generation of BSW and RTE
**AUTOSAR Workflow (theory)**

Complete SW functionality of the vehicle is defined as a system of SWCs...

An extract is created for each ECU...

... and distributed to ECUs

The ECU is configured based on the ECU Extract.

Extract of System Description*

System Description*

Basic Software (BSW)

ECU Configuration Description (ECUC)*
Typical Scenario

- Various different tools are involved in development of ECU software
- SWCs need to be integrated from different sources, developed with different technologies
Interaction with ModelingTools

Workflows

Input Files (from OEM)
- Legacy Formats
  - .ldf
  - .fibex
  - .dbc
- System Extract / Description
  - .cdd
  - .odx
  - .arxml
- Diagnostic Data
  - .arxml

Model-based development tools

Legacy File Conversion
- .arxml

Extract Generation
- .arxml

Extract Merge
- .arxml

AUTOSAR ECU Development Tools

DaVinci Configurator Pro

DaVinci Developer

MICROSAR RTE files

BSW configuration files

Code generation (requires MICROsAR license)

SWC implementation files (generated)

SWC implementation files (legacy, hand-coded)

Leaf Files
- .c/.h
- .a2l

SW Component Description
## AtomicComponent implementation

- Model-Based Design tools like Simulink or TargetLink can be used to generate the component implementation.
- Usually an ECU Application consists of a mixture of generated and hand-coded SWC Implementations.
- An AR-Design needs all the information to generate and optimize the RTE.
  - Besides that AR is not interested in the internals of an SWC.
  - The RTE is indifferent in which way the SWC is implemented.
- The specific code generator requires an according AR target to produce APIs according the Standard.

### Diagram:

```
+---------------------------------+                +---------------------------------+
| Model-based development tools   |                  | SWC implementation (generated) |
| Model-based development tools   | arxml            | SW Component Description      |
| AUTOSAR ECU Development Tools   | DaVinci Developer| SWC implementation (generated) |
| Templates                       |                  | SWC implementation (legacy,    |
|                                 |                  | hand-coded)                   |
```

- DaVinci Developer is used for the implementation of SW Components.
- Model-based tools provide the toolchain for model generation.
- Model-based tools provide APIs according the Standard.
SWC Integration

**DaVinci Developer**
- Create SWC descriptions
- Generate SWC header files and implementation templates
- Integrate SWC descriptions into System Extract

**DaVinci Configurator Pro**
- Create ECU Extract based on System Extract (flattening)
- Integrate SWC descriptions into ECU Extract
- Create service SWC descriptions
- Configure RTE (task mapping, ...)

**Model-based development tool**
- SWC description generation
- SWC integration

**IDE**
- SWC compilation and testing

**System Extract**
- Comp1
- SWC3
- SWC4

**ECU Extract**
- SWC2
- SWC3
- SWC4
- Service SWC1
Interaction with ModelingTools

Workflow Scenario

- **Main Scenario**
  - Re-use existing Models
  - SWC Interfaces can be designed according the model requirements
- **Clarify**
  - The AUTOSAR- and tool-versions to use
  - How to transfer the model to get the desired structure in AUTOSAR
  - DaVinci supports and understands the used AR Features
    > Vector together with given MbD-Tool vendors is constantly justifying this for typical use-cases
Workflow Scenario

Main Scenario
- The complete SWC is defined by the design in DaVinci
- Use the model creation/import to realize the design as model
- Helpful if you have a clear understanding or the AR design of your SWC

Clarify
- The AUTOSAR- and tool-versions to use
- Clarify exactly which features are supported, by tool and import
- The created code needs to match the imported SWC in every detail
- Is the iterative import supported correctly
Workflow Scenario

- **Main Scenario**
  - Define the SWC interfaces by the AR design
  - Let the Model define the SWC InternalBehavior

- **Clarify**
  - All of the first two Scenarios
  - Is the multiple iterative roundtrip supported correctly
    - Clarify the handling of PackageStructure
  - Ensure to keep the Import/Export options consistent
<table>
<thead>
<tr>
<th></th>
<th>Agenda</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Vector AUTOSAR Tool Chain</td>
</tr>
<tr>
<td>2.</td>
<td>Interaction with ModelingTools</td>
</tr>
<tr>
<td>3.</td>
<td>Tool Versions and Features</td>
</tr>
<tr>
<td>4.</td>
<td>Summary and Demo</td>
</tr>
</tbody>
</table>
AUTOSAR Support is part of the Simulink EmbeddedCoder
The AUTOSAR Target is one of the supported platforms
The currently recommended Version is the new Release 2014b
In general from Release 2013a AR version 4.x is supported
   It was recommended to install the ATPP – AddOn
For a detailed AR version support please contact MathWorks directly

Separate AR configuration dialog for the specific AR settings,
no direct model annotations necessary
The EmbeddedCoder generates one Subsystem as SoftwareComponent
with possibly several runnables
dSpace TargetLink

- TargetLink 2.3.1 already supported AR version 3.x
- TargetLink 3.3 supported also AR version 4.x
- For a detailed overview of supported AR versions and features please contact dSpace directly

- Detailed AR settings supported by the DataDictionary
- AR specific block-set that allows the required model annotation of AR settings
  - Runnables are modeled as triggered subsystems
  - Components are instantiated and visible within the DataDictionary
  - A runnable InPort can be a S/R or a C/S communication or an exchange via IRV
Tool Versions and Features

Application DataType

- Abstract data type definition, conceptual/algorithmic view
  - Used for modelling purpose
- CompuMethod describe the physical representation
- Can be used in Port Interfaces
- Dedicated mapping to an Implementation Data Type required for RTE generation
  - Application DataTypes will show up in the generated RTE code.
Tool Versions and Features

Implementation DataType

- Concrete data type, implementation view
- CompuMethod describes additional properties like enumerator values
- Can be used in PortInterfaces
- References the specific base type
  - Recommended to use a DataRef to a platform type directly on implementation level
Tool Versions and Features

BaseType

- Describes code-level type,
- no semantics, no direct PortInterfaces usage
- Native Declaration describes the code symbol
  - Should be specified only if the Implementation DataTypes was not already a platform type.
DaVinci Developer Import Features

- **Package Import**
  - Defines the package structure of the imported elements
  - Usually import from file is the recommended choice
  - Ensure to use the same setting for every import of an SWC

- **Substitute missing Objects**
  - Allows to resolve open references to already existing Wsp-elements

- **Lock created objects**
  - Helps to prevent modifications of imported elements

- **Default import settings**
  - can be specified under workspace settings
Agenda

1. Vector AUTOSAR Tool Chain

2. Interaction with ModelingTools

3. Tool Versions and Features

4. Summary and Demo
Summary and Demo

Demo
Summary and Demo

Summary

- The AUTOSAR specification itself allows an interaction with MbD-Tools
  - Given that AUTOSAR as platform for code generation is supported
  - The tools support according flexible import / export scenarios

- The very flexible DataType model of AUTOSAR is a challenge
  - MbD-Tools need to support this for simulation and code-generation
  - Usually there are some restriction e.g. regarding pointers or multi-dimension complex types

- The persistent handling of AR-Structures like Packages and UUIDs
  - Here at least a DaVinci DEV Version 3.6 is recommended

- Large number of Tool and AUTOSAR Versions in different combinations makes a general recommendation quite difficult
  - If in doubt please contact the tool supplier for your specific project or problem
For more information about Vector and our products please visit

www.vector.com

Author:
Peter Hajsani
Vector Informatik GmbH