Electric and Safe

Using modeling guidelines to validate e-drive software

Together with the TargetLink Strategic Partner Model Engineering Solutions, Mercedes-Benz Research & Development North America developed automatically testable conformity rules for modeling with dSPACE TargetLink, which comply with important requirements of the ISO 26262 standard.

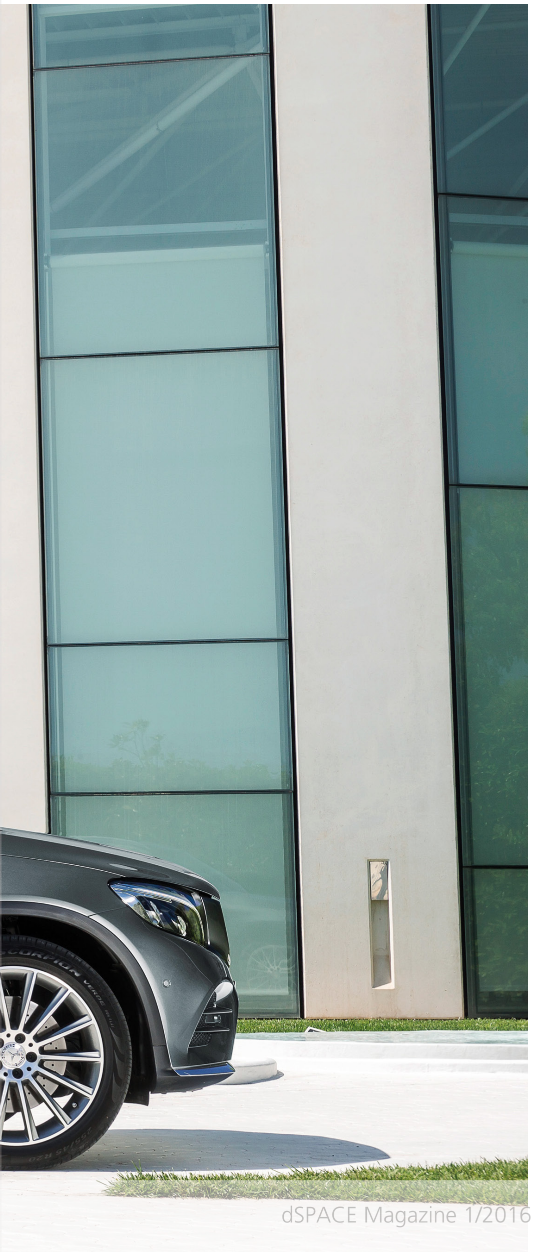
Mercedes-Benz Research & Development North America (MBRDNA), with a location in Redford, Michigan, amongst others, is responsible for developing and integrating inverter software (current conversion for electric machines). This software, developed with dSPACE TargetLink®, is used for various vehicle applications in the Mercedes-Benz e-drive portfolio. One main component, and the actual innovation, is the e-motor control and associated torque and high-voltage safety concept that is implemented according to the ASIL C requirements of ISO 26262.

Development Process and Model Quality

The e-drive software is based on the AUTOSAR software architecture and

Source: © Mercedes-Benz

can be scaled for many electrified drivetrains. The complete control software is developed with the model-based design method according to the V-cycle. TargetLink, dSPACE's production code generator, is a core component of the development tool chain. TargetLink supports modeling and code generation of an AUTOSAR-compliant software architecture and is certified up to ASIL D for safety-related software. Modeling the function software in Simulink® and TargetLink plays a central role for early requirement validation, because a higher quality of the models used for code generation directly translates into a higher quality of the generated software. Using Simulink/TargetLink for software modeling is an accepted, industry-proven method to generate high-quality software. This is also in line with ISO 26262, which recommends using a semi-formal modeling language such as Simulink. MBRDNA uses a combination of static and analytical validation measures to ensure a high model quality. Independently of the functional customer requirements, MBRDNA defined development methods that ensure an optimal integration of the software into the target environment.



Rules for the Software Design

An important part of this method is the consistent use of modeling and conformity rules for software design. The rules used at MBRDNA are based on Daimler-internal regulations for model development and have been adapted to the development requirements of the e-drive software. The Daimler modeling guidelines are based on modeling standards and tool-specific guidelines such as MAAB, MISRA Simulink/Stateflow, MES Functional Safety Guidelines, MISRA TargetLink, and the dSPACE TargetLink Modeling Guidelines (figure 1). Because all of these guidelines have a different focus, a smart combination of them is needed to cover all aspects required for modeling safety-related software according to ISO 26262. The rules of the MAAB (MathWorks Automotive Advisory Board) focus on design aspects of simulation and controller models with an emphasis on readability, serviceability, and best practices. The MAAB rules do not accentuate production code generation. The MISRA Simulink/Stateflow and MISRA TargetLink guidelines, however, focus on safety aspects of the models and the code generated from them. They define a safe language range for Simulink and Stateflow, modeling patterns for safe code patterns, and an appropriate configuration of the simulation environment. Tool-specific guidelines such as the MISRA/TargetLink and dSPACE TargetLink Modeling Guidelines predominantly refer to code generation with TargetLink. Compliance with these guidelines means that there must not be any modeling patterns or configurations of the model or code generator that can negatively affect the properties of the generated code. The MES Functional Safety Guidelines largely refer to safety considerations of the model and the generated code. These guidelines were derived from the requirements of ISO 26262 and other safety standards, and complement the existing guidelines for the design of safety-related software. Key elements of the analyses are checks for data flow and control flow.

Automated Testing for TargetLink Models

With its goal of making it easier to use the modeling guidelines, MBRDNA tailored the Daimler modeling guide-

>>

> "The increasing complexity of software systems pushes traditional testing to its limits. Automated analyses of the created models and the translated software are an integral part of our software quality assurance."
>
> *Alexander Dolpp, Mercedes-Benz Research & Development North America, Inc.*

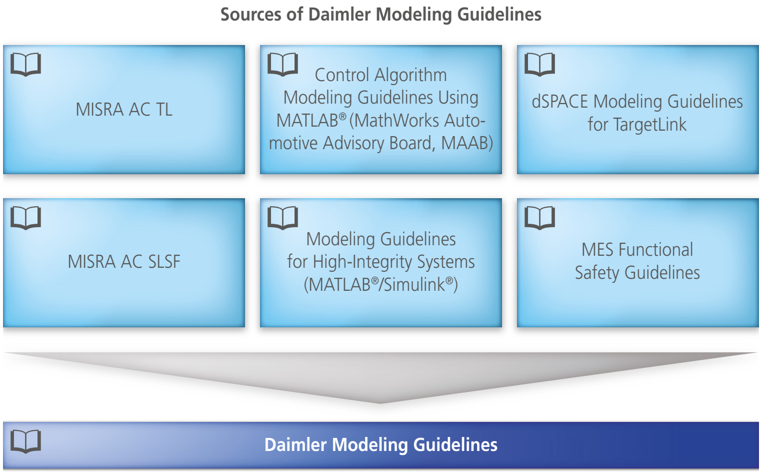

Sources of Daimler Modeling Guidelines
MISRA AC TL
Control Algorithm Modeling Guidelines Using MATLAB® (MathWorks Automotive Advisory Board, MAAB)
dSPACE Modeling Guidelines for TargetLink
MISRA AC SLSF
Modeling Guidelines for High-Integrity Systems (MATLAB®/Simulink®)
MES Functional Safety Guidelines
Daimler Modeling Guidelines

Figure 1: The Daimler modeling guidelines are based on numerous established standards and guidelines.

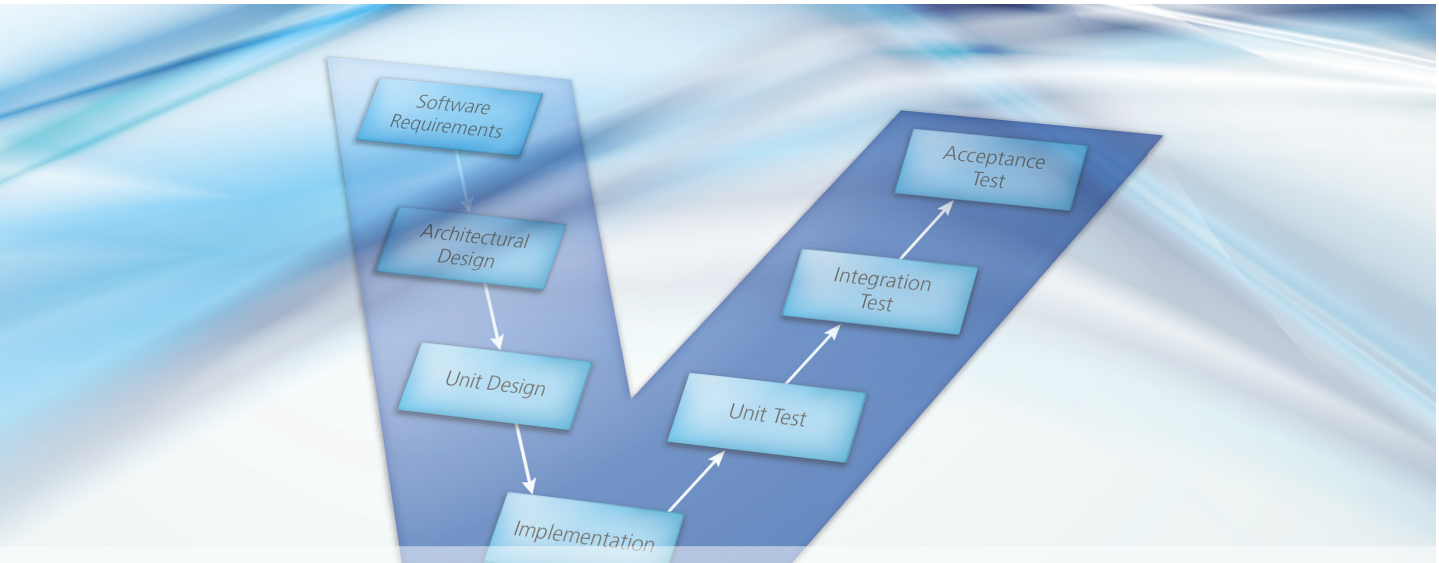
lines together with Model Engineering Solutions GmbH (MES), a TargetLink Strategic Partner, on the basis of the existing documents, and extended them to cover Daimler-specific needs. The guidelines are managed in MES Model Examiner (MXAM) and automated tests for TargetLink models are provided. By adhering to the modeling guidelines, MBRDNA can comply with the modeling requirements of ISO 26262, implement best practices, avoid modeling mistakes, and consider tool-specific configurations, such as uniform settings in the simulation environment or the code generator.

Safety in Detail

For example, as an adapted MISRA modeling guideline, the Daimler modeling guidelines determine the correct execution order of Stateflow transitions. Here, the Daimler guidelines contradict the MISRA guidelines by stating that the execution order should be defined only by the user, not by the graphical order of states and transitions, to avoid a misinterpretation of the Stateflow semantics. If, for instance, a state is moved for layout reasons (change in syntax), this can inadvertently change the execution behavior of the state (change in semantics), see figure 2a. The same holds true for Stateflow transitions that are evaluated at junctions, as shown in figure 2b. This misinterpretation or unintended

> “Using modeling guidelines and a guideline checker like MES Model Examiner makes it possible to automatically implement the requirements of ISO 26262 and give the model experts more freedom. We want them to focus on their main task: developing the control function.“

Dr. Ingo Stürmer, Model Engineering Solutions



> "With its native AUTOSAR support, TargetLink for code generation is a core element of our development tool chain."
>
> *Alexander Dolpp, Mercedes-Benz Research & Development North America, Inc.*

change of the execution behavior can easily be avoided by using the Stateflow configuration "User specified state/transition execution order" to calculate the execution order as specified by the modeling expert. Model checking in MES Model Examiner (MXAM) is used to automatically verify whether a model complies with these guidelines and to correct the model immediately.

Clear Processes

All modeling experts and software developers must adhere to the modeling guidelines. Automatic model checking with MXAM has to be performed each time a new functionality is added. Software can be checked into a version management system only if it complies with the modeling guidelines and the associated functional model-in-the-loop (MIL) tests were performed. The modeling expert is responsible for eliminating all violations that were identified on the basis of the guidelines. This static modeling analysis method is part of all validation measures in the V-cycle for developing and validating the e-drive software. By using the Daimler modeling guidelines together with MXAM for automatic conformity checks and the dSPACE production code generator TargetLink, Mercedes-Benz Research & Development North America uses an industry-proven approach to comply with ISO 26262 requirements, improve model quality early on and significantly increase code quality. ■

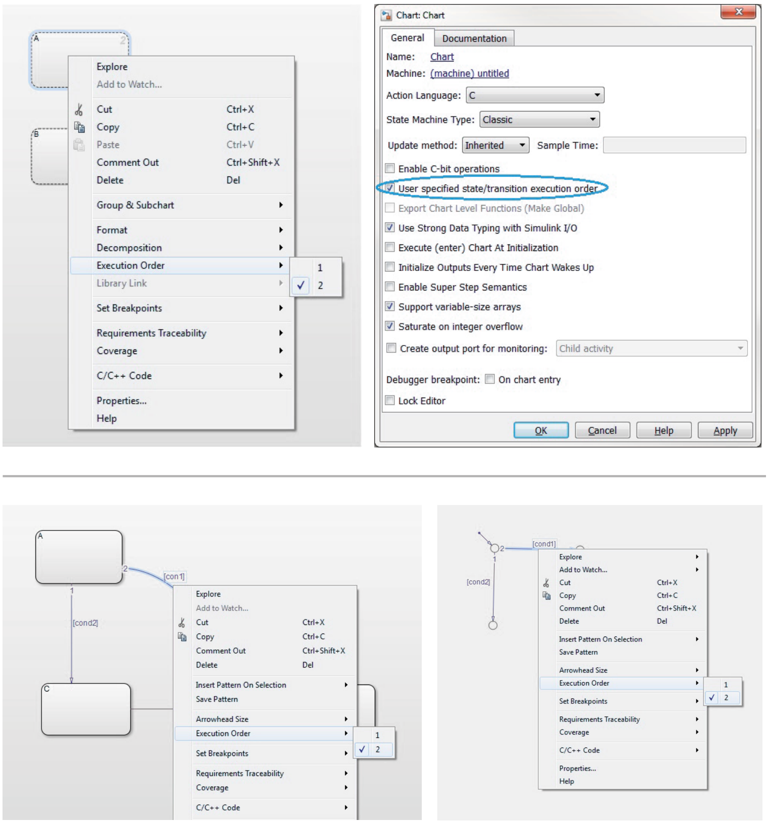


Chart: Chart

General | Documentation

Name: Chart
Machine: (machine) untitled
Action Language: C
State Machine Type: Classic
Update method: Inherited   Sample Time:
☐ Enable C-bit operations
☑ User specified state/transition execution order
☐ Export Chart Level Functions (Make Global)
☑ Use Strong Data Typing with Simulink I/O
☐ Execute (enter) Chart At Initialization
☐ Initialize Outputs Every Time Chart Wakes Up
☐ Enable Super Step Semantics
☑ Support variable-size arrays
☑ Saturate on integer overflow
☐ Create output port for monitoring: Child activity
Debugger breakpoint: ☐ On chart entry
☐ Lock Editor

Figure 2a (top): A typical case for MES Model Examiner – Execution order of parallel states, here specified in Stateflow.
Figure 2b (bottom): Execution order of transitions.

Alexander Dolpp
Alexander Dolpp is Director for E-Drive Software at Mercedes-Benz Research & Development North America, Inc., Redford, Michigan, USA.

Dr. Ingo Stürmer
Dr. Ingo Stürmer is the founder and former CEO of Model Engineering Solutions GmbH. Since January 2016, Dr. Stürmer heads Model Engineering Solutions Ltd. (UK).