

Mercedes-Benz R&D North America: Enhanced Safety and Efficiency for the Software Design Process at Mercedes-Benz

How Mercedes-Benz Research & Development North America (MBRDNA) uses model metrics from MES* to boost the model-based software design process.

As the research and development arm of one of the world's most renowned car manufacturers, MBRDNA's development team is constantly looking to implement the most efficient and effective methods and processes to support the development of safe software for cutting-edge automotive functionality such as E-Drive controllers. However, adding functionality to an existing software program while keeping it understandable, maintainable, and testable is a challenge. Advanced tools and processes are required to meet this goal, even when using software models.

Continuous improvement of control software: The need for refactoring

Plug-In Hybrid vehicle family (Source: MBRDNA)

Plug-In Hybrid vehicle family (Source: MBRDNA)

In particular, MBRDNA was looking to enhance the capabilities of the control software for their E-Drive components. The team, located in Redford, MI (USA), has already adopted model-based development as its key development approach for embedded control software design. As part of this, the software model serves as the fundamental artifact in the development process – the functional integrity and the coverage of the intended functionality are primarily verified at the software model level. The existing software models used for production are periodically enhanced to incorporate more and more functionality as part of a continuous improvement process. The E-Drive software model in question at MBRDNA serves as the basis for the controller of an electric drive system. It manages functions such as torque and traction of the electric drive. This controller software is also used for 48V systems, pure E-Drive systems, and fuel cell programs.

The MBRDNA strategy was to first implement the basic functionality for the software. As more complex requirements and additional project stakeholders, such as new developers, testers, and calibrators, got involved, MBRDNA identified the need to “manage complexity” in order to keep the complex software model relatively easy to understand and promote straightforward communication about it among the team. The software model had grown significantly, and this became the main driving force behind reviewing the basic model. The software models' functional testing team, in particular, was often required to invest extra effort in testing the software module and the units involved.

Developing a systematic approach for software model refactoring

MBRDNA first defined a set of objectives for refactoring software models. **The main objective was to improve the testability and understandability of the model by reducing the complexity.** MBRDNA was looking to reduce both subsystem-level complexity and also complexity at the level of the entire

integrated software model. MBRDNA complies with the ISO 26262 safety standards, so it is essential for them to ensure that the units and models are well structured and do not contain overly complex parts, as these frequently cause issues in testing and in general. Overly complex parts have to be split up into less complex and smaller units – ideally containing one dedicated functionality per subsystem. An additional objective is to identify potential library elements, as this will further reduce overall model complexity. MBRDNA's approach is to drive automation to the highest level. In line with this, MBRDNA set about looking for a model architecture enhancement tool to support and monitor the refactoring results. The long-term intention was to verify if such a tool can help support continuous architecture monitoring and optimization as part of the defined software model design process.

Finding the right tool for the job

To support the model refactoring project, model metrics from MES* were selected. These compute the local complexity of software models for each subsystem and also deliver the global model complexity. This function also points the user to so-called hotspots of the model. Hotspots are subsystems of the model that should be revised and refactored. To further optimize impact of this function, MBRDNA took advantage of their flexibility, customizing them according to their specific environment. In particular, MBRDNA added specific function blocks and defined custom scaling factors to the configuration of the MES model metrics*. Due to this easy configurability, a wide range of metrics could be delivered, identifying problematic parts of the MBRDNA models accurately. This function provides both a qualitative and a quantitative analysis. The quantitative analysis includes the overall global complexity of the software module. This information is needed in the “re-architecting process” and for the further modularization of software, where global complexity is used to determine which oversized software module should be split into several software modules. The qualitative analysis provided MBRDNA information about error-prone hotspots within a particular software module. Here, MBRDNA is specifically interested in the local subsystem complexity and how to limit complexity in a smart way.

Remodeling techniques applied

Based on the metrics delivered by MES tools, MBRDNA developed and employed a range of techniques to refactor the identified software model parts:

- Systematic splitting up of overly complex subsystems into a set of less complex, functionally coherent subsystems.
- Use of the MXAM/MXRAY* clone detection to detect redundant elements/blocks in the model (e.g. blocks for Polynoms) that are used repetitively. These redundant elements are then added to the MBRDNA specific block library or to a referenced model. The basic idea here is to avoid redundant model components that have to be reviewed and tested individually.

Yielding positive results with MES model metrics*

Humphrey Achiri Senior Developer with MBRDNA
Humphrey Achiri,
Senior Developer with
MBRDNA

The analysis using model metrics from MXAM/MXRAY* indicates error-prone hotspots within a software module with high accuracy. According to Humphrey Achiri, Senior Developer at MBRDNA, “**This considerably improved the overall readability, testability, and maintainability of our software modules.**” Beyond this, the calibration engineers at Mercedes-Benz pointed out that it “has become much **easier to navigate** through the finalized software during vehicle testing, where time on the test tracks is a constraint.” It has also become more convenient for the Mercedes-Benz test engineers to set up MIL and SIL test cases (**i.e. easier to perform requirements-based testing**) since individual

requirements were better aligned with the actual implementation in particular subsystems. Additionally, it has become **easier to document** the software, thanks to the less complex individual subsystems.

MBRDNA has been able to achieve its main objective of improving the readability, testability, and maintainability of its software modules for the new generation of hybrid and fully electrical cars. Documentation of the software has also been simplified due to the reduction in complex individual subsystems. The numeric results achieved during refactoring include:

- Global complexity, a metric for the understandability and readability of the entire software model, has been **reduced by around 10%**. This signifies a substantial reduction in the effort required to understand, maintain, test, and implement the refactored software model.
- **A substantial reduction in local complexity values was achieved**, and for some software modules, the number of complex subsystems was reduced to zero.

Results achieved in soft facts:

- Comments from the requirements engineering team: **“Wow – these software models are much easier to understand and to work with now.”** With the model’s additional relevance to MBRDNA’s 48V systems, pure E-Drive systems, and fuel cell programs, managing its complexity has also proven beneficial for variant management.

Additionally, the automatically generated code has not changed, a result of the fact that the functionality described in the model was not modified.

In light of the positive tangible results, the process improvements, and the favorable internal feedback, the effort MBRDNA dedicated to setting up MES tools and to improving the entire software model has been very well invested. The feedback and the actual concrete outcomes of the refactoring project are very good.

Outlook on complexity management at MBRDNA

Alexander Dolpp, Director E-Drive software at MBRDNA
Alexander Dolpp,
Director E-Drive software
at MBRDNA

In the future, the model metrics from MXAM^{*} will support continuous model optimization and refactoring at MBRDNA. Using MES tools, the process guidelines for developing and refactoring large-scale software models for controller functions at MBRDNA will lay out a set of checks and metrics as a team-wide best practice to monitor the model architecture across the entire development process. MBRDNA also plans to integrate the tool into its highly automated Continuous Integration build process, which is based on Jenkins technology. With additional MES model metrics, such as incoherence and interface efficiency metrics, “the process for continuous model enhancement can be streamlined even further” concludes Alexander Dolpp, Director E-Drive software at MBRDNA.

* Model metrics were originally part of MES M-XRAY (MXRAY)[®]. The MXRAY features have since been integrated in to MES Model Examiner[®] (MXAM) where they are continuously evolving.