New paths in software engineering

The rapid growth of complexity of software-controlled vehicle systems is pushing current workflows between function and software developers, calibration engineers, and testers to their limits. This is where two new ETAS tools come in.

ETAS has developed two new tools for understanding, describing, and optimizing complex systems.
SCODE-ANALYZER uses a relationship matrix that enables developers to understand even the most complicated interdependencies.

From low-emission powertrains to dynamic chassis to self-driving cars, software is both an enabler and a guarantor of a high level of reliability. What makes life easier for drivers poses major challenges for developers: they need to have a clear view to cleanly implement functions in increasingly connected vehicle systems despite growing system complexity - not to mention the pressure to develop reliable and functionally secure software at a low cost and within a short space of time.

This is where virtualization can help – but to implement complex systems correctly, developers must already have an understanding of their interdependencies and limits, even when these are still very abstract. Using simulations in the function development process can help, but these are of little use if artifacts are incomplete or invalid. The model-based approach, on the other hand, is the most efficient path to functionally safe ECU code if the assumptions and system descriptions are correct right from the start.

New method for model-based software development

ETAS has developed two new tools for validating system descriptions: SCODE-ANALYZER and SCODE-CONGRA. Both introduce a completely new model-based software development method based on functional morphology, enabling software developers to verify semantic descriptions of a software's decision paths and their mathematical relationships automatically.

In SCODE-ANALYZER, developers enter the decision paths of the planned system as defined in the specifications, either as a graph or a table. The tool automatically checks the plausibility of the description. The described system is broken down into “Modi” and the tool compares the overall system with these subsections to check its integrity and identify any unnecessary sections.

This structured system description provides a clear overview of highly complex systems on its own, but the automated check is required in order for SCODE-ANALYZER to automatically derive program code from the semantic or graphic descriptions. Developers can choose to have the tool generate artifacts in the form of m-files for MATLAB®, s-function blocks for Simulink®, ESDL code for ASCET-DEVELOPER, or C code. One of the USPs of SCODE-ANALYZER is that the tool provides proof that all input conditions have been considered, which is a major advantage particularly when it comes to safety-critical systems.

From exact formulas to secure code

SCODE-ANALYZER comprehensively secures model-based function development. In addition, the basic data is identical for the simulation code and the production code for ECUs, guaranteeing a high level of correlation between the simulation model and the production code – and thus a high degree of functional safety of the ECU software.

In SCODE-CONGRA, function developers use physical formulas from the literature to describe systems. The tool then derives undirected or directed graphs and displays them in an interactive visualization. Developers can change values in these systems and see how these changes influence system behavior, which brings complex interdependencies to light as well as any lack of clarity or consistency of the input data.

The visual representation facilitates early sensitivity analyses, which helps with the subsequent calibration. Calibration engineers see right away which adjustments influence the system functions and whether they result in unexpected or undesired side effects. The ability to translate common formulas from the literature into this interactive visualization makes it easy for
The new tools create transparency in model-based application development. The automated plausibility checks of all inputs give even inexperienced developers security in designing and implementing complex safety-relevant functions, and thoroughly validated software, whether it controls modern powertrain and chassis systems or autonomous vehicles.

**Summary**

Precise formulas in SCODE-CONGRA or descriptions in table and graphic format in SCODE-ANALYZER become the basis of automated software development. Function developers benefit from the option to use accurate simulations at an early stage, and calibration engineers benefit from reliable sensitivity analyses. ESDL code that is already validated facilitates the work of software developers. And test engineers will soon be testing software that already has a high level of maturity. In short, ETAS’ new method based on functional morphology paves the way for more efficient workflows in software engineering.

**Authors**

Dr. Markus Behle is a senior product manager for SCODE-ANALYZER and SCODE-CONGRA at ETAS in Stuttgart, Germany. Alexander Mayer is a field application engineer at ETAS in Ann Arbor, Michigan, USA.