In the modern vehicle cockpit, Michael Beine and Olaf Grajetzky discuss what it means for TargetLink if functional requirements increase and development cycles become shorter and shorter.

20 Years of TargetLink

TargetLink has been generating highly efficient code for series production quickly and reliably for 20 years. Today, TargetLink-generated code is used everywhere in the vehicle, and TargetLink is an integral part of automotive software development worldwide. Impressive use cases can also be found in other industries. We discussed the role of TargetLink with Michael Beine, Lead Product Manager, and Olaf Grajetzky, Group Manager Engineering at dSPACE. Both have contributed to the success story of TargetLink.

From highly efficient production code generator to agile model-based software development

Validation on Real-Time Hardware
TargetLink now makes testing production code on SCALEXIO hardware even easier by using-Simulink implementation containers (SIC). Learn more about the benefits of this option on page 49.

Mr. Beine, what was the greatest benefit for our customers when they introduced TargetLink?
Michael Beine: From the very beginning, TargetLink was more than a new software product. TargetLink closed the gap between the function model and the ECU. Because our customers were able to switch from manual programming to automatic code generation, they were able to significantly accelerate their development process. Before TargetLink was introduced, the turnaround time from design to implementation was weeks or even months. After the initial configuration, TargetLink generated code at the click of a button. And in addition to the considerable increase in efficiency, our customers were able to further improve software quality assurance. This was possible because code and models are consistent and the process for comparing them using MIL, SIL and PIL simulation is simple. The first projects for which Nissan and MAN used TargetLink were something of a revolution.

When you look back at the beginnings of TargetLink: What has changed since then?
Michael Beine: Initially, the focus was clearly on code generation and primarily on code efficiency. Today, it is very important to enable teams and entire departments to develop and validate model-based software efficiently and safely. With this in mind, we continuously developed TargetLink and built a comprehensive software ecosystem that addresses model-based software development, including validation.

What were the important development steps?
Michael Beine: From the very beginning, weworked closely with many customers in their projects. This, for example, lead to creating the TargetLink Data Dictionary. For the first time, it was possible to disconnect the implementation details from the model and exchange them within the team and across teams. Moreover, we were and are active pioneers in the field of AUTOSAR. In 2006, TargetLink was the first code generator to support AUTOSAR. Since then the tool has offered unparalleled AUTOSAR support at product level. We are currently working on Adaptive AUTOSAR. Not to forget: Since 2009, TargetLink has been officially certified by TÜV Süd for use in safety-critical projects.

Speaking of safety. How much safety does TargetLink offer?
Michael Beine: Developers and engineers are on the safe side when developing software with TargetLink. An important point of orientation for our customers is a reference workflow that is part of the TÜV approval. This workflow provides guidance for model-based development of safety-related software using TargetLink. In this context, our two strategic partners, Model Engineering Solutions and BTC Embedded Systems, play an important role. Their tools in conjunction with TargetLink ensure the required quality

>>

at the model and code levels. Modeling guidelines, MISRA compliance, dedicated support by dSPACE experts, regular Automotive SPICE development audits, and patches for earlier TargetLink versions are further aspects that show how important quality and reliability are to us.

Mr. Grajetzky, in which areas is TargetLink used, and which use cases do you particularly remember?
Olaf Grajetzky: Production code that was generated with TargetLink can be found in any ECU in a car, for example, in the powertrain, chassis, body, or ADAS areas. Morevoer, TargetLink has also proven itself in many other industries. For example, in motor controls for angle grinders, where you would hardly suspect automatically generated production code. Developments for self-propelled harvesters are quite impressive, and it is remarkable that even satellites in orbit are equipped with control systems that contain TargetLink code.

dSPACE is on site to implement many projects. How does the user support look like?
Olaf Grajetzky: One example is a major automotive customer, for whom we have been supporting a highly automated tool chain for more than ten years. This tool chain now supports all relevant specifications of AUTOSAR 4. In my opinion, this is one of the most consistent and sophisticated applications of AUTOSAR. In many other projects, we provide our know-how only for short-term support at the start of the project. Our customers then take over the operation. For almost 20 years, we have been on site with our customers and can quickly find solutions most suitable for their requirements. Even if problems arise, we can help at short notice. This is important because requirements can be very different around the globe, therefore we often have to be rather flexible.

Especially in the automotive sector, functional requirements are increasing and development cycles are becoming shorter. What does this mean for the continued development of TargetLink?
Michael Beine: We are currently working on implementing Adaptive AUTOSAR, another standard that, among other things, allows for updating ECUs and thus advances function development for autonomous driving. TargetLink will support requirements that involve new technologies, such as continuous integration and agile methods, and help our customers further increase their development speed. Generally, it is important to continue on the path towards agile, model-based software development and to ensure smooth interaction throughout the entire tool chain.

What role will TargetLink play once the vision of autonomous driving becomes a reality?
Olaf Grajetzky: One important role will remain unchanged: Production code generated with TargetLink will always be required for vehicles to drive safely on the road. Because code that is used to steer, accelerate, or brake vehicles not only in critical situations has to be absolutely secure, deterministic, and reliable.

Mr. Beine, Mr. Grajetzky, thank you for talking to us.

Michael Beine is Lead Product Manager at dSPACE.

Olaf Grajetzky is Group Manager Engineering at dSPACE.

Validation on Real-Time Hardware

Let's take a closer look at one of the latest developments in TargetLink: As of Version 4.4 (dSPACE Release 2018-B), TargetLink offers the option of exporting production code directly from TargetLink as Simulink implementation containers (SIC), executing it on the dSPACE SCALEXIO real-time hardware via ConfigurationDesk, and thus validating the code quickly and particularly conveniently. Product Manager Felix Engel explains the benefits of this additional validation step.

Mr. Engel, which users and applications does the new feature address?
First of all, the feature is aimed at software developers. They can actively work with the real production code generated by TargetLink, which has already been tested using model-in-the-loop, software-in-the-loop, and processor-in-the-loop simulation, early in the development process with real-time hardware on a real controlled system. This allows developers to directly see the effect that the specific implementation of an algorithm as production code has on the function – taking into account resource restrictions and quantization effects of all kinds. We also address test engineers who want to systematically test functionality that is available as a TargetLink model with real-time hardware on the real controlled system. Thanks to the clear interface of the SIC container, our implementation software ConfigurationDesk now makes this workflow easy to use and also ensures the highest degree of process reliability because function and I/O are separated. As a result, relevant file versions, for example, can be clearly assigned.

What is the benefit of this validation method compared to the well-known MIL/SIL/PIL simulations?
The methods complement each other. The MIL/PIL/SIL simulations allow for testing a wide variety of variants. If required, they let you easily compute high data volumes in parallel in large clusters. As a result, they can cover a great variety of tests in all dimensions. The validation of the production code on the real controlled system adds the element of early random checks for simulation plausibility to the tests. The objective is to notice effects that are not visible during the simulation. These effects can be detected very early on and thus remedied easily.

Which effects can go unnoticed in simulations?
In addition to the quantization effects mentioned above, an environment model might not represent the environment accurately enough, for example. In reality, however, the function modified for production code use must prove that it can cope with the complex details of the environment. Executing production code on SCALEXIO hardware leads to quick answers.

Which hardware is used?
You can use all ConfigurationDesk-supported systems. The SCALEXIO LabBox and, above all, the new SCALEXIO AutoBox (see interview on page 40), which were both designed as prototyping systems, are particularly suitable for validating production code.

What does the additional validation step mean for the overall ECU validation process?
It further increases confidence in the validation of production code and the later control unit. In addition, the overall costs are reduced, because more and more risks are identified and eliminated in advance.

Mr. Engel, thank you for talking to us.

Felix Engel is Product Manager at dSPACE.