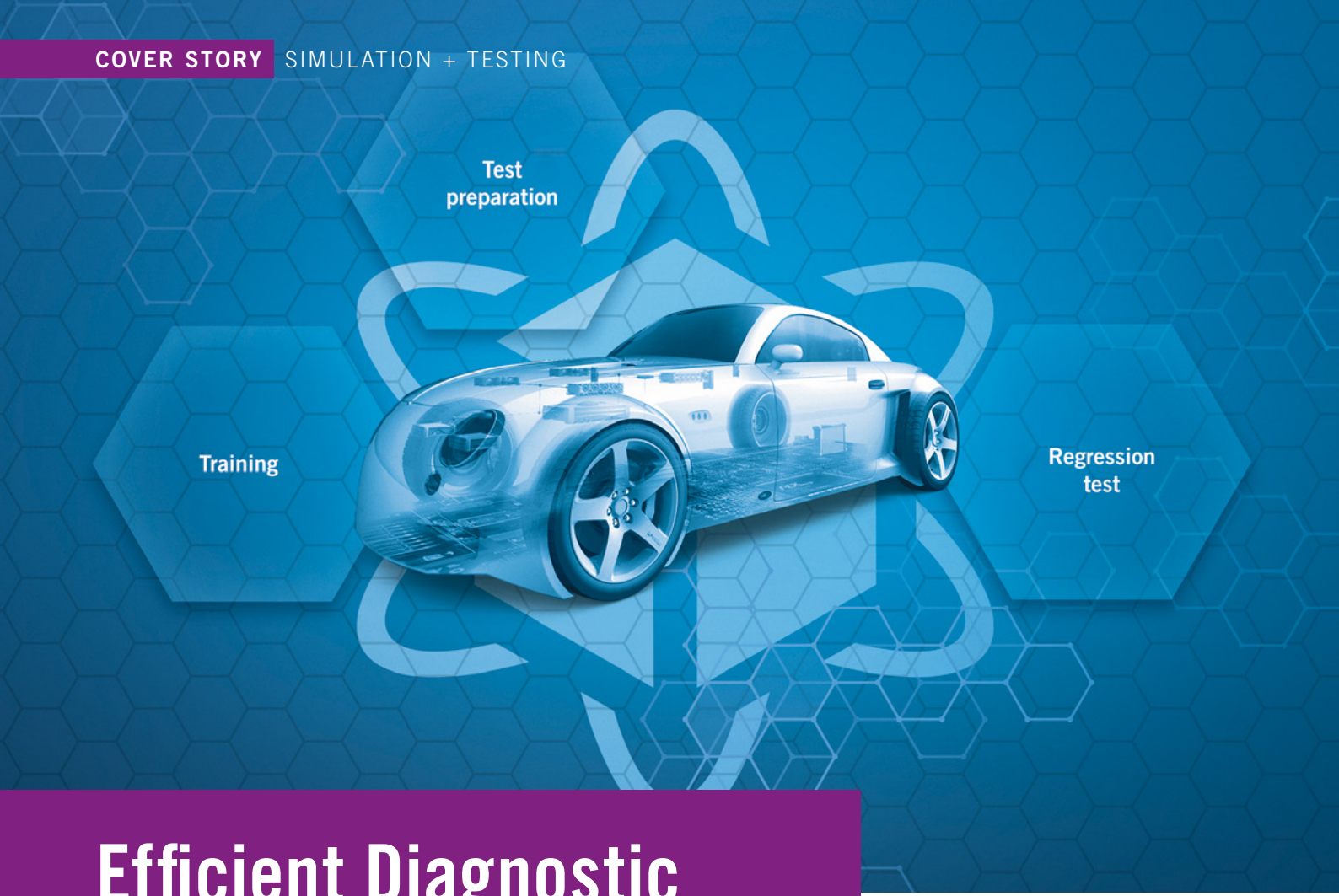
Springer Vieweg

# ATZ extra


SIMULATION + TESTING

# Using Simulation to Represent Reality

optimize!
softing




© Softing

# Efficient Diagnostic Testing with Simulation

Testing and diagnosing modern vehicles is becoming increasingly complex. The evaluation of the data recorded during operation plays an important role here, as it forms the basis for diagnosing any errors that may occur. In this article, Softing shows how simulation methods can be used efficiently for the use of this data and the preparation of testing.

AUTHOR

**Markus Steffelbauer**
is Director Product Management at Softing in Haar near Munich (Germany).

Modern vehicles are computer-based networks on wheels. Driving and convenience functions are increasingly distributed across several Electronic Control Units (ECUs), with most functions still consisting of a mechanical component and the computer component. Testing this complex interaction of mechatronic components is essential, not least for safety reasons – and is both an expensive and complex undertaking.

Diagnostics plays a very special part in it all. During operation, diagnostics is responsible for permanently monitoring the operating states of the mechatronic system and thus represents a vehicle function in its own right. At the same time, it makes the results of its checks available to external devices as required, making it indispensable for the vehicle manufacturer in manufacturing and the repair shop. However, it is thus also part of the test methodology. For diagnostic tests, this results in a dichotomy: on the one hand, testing the diagnosis, and on the other hand, testing using the diagnosis.

The term "diagnostics" describes both ECU-internal algorithms, which are used to determine errors in sensors/actuators and ECU communication, and the transmission protocols to an external diagnostic tester. Both cases have to be validated. This necessitates, for example, the creation of error conditions, which are subsequently checked to see whether an error has been cor-


**FIGURE 1** Simulation in test preparation (© Softing)

rectly detected and entered in the error memory. With regard to the protocols, transmission characteristics and data content are to be tested.

## VALIDATION OF TESTING FUNCIONALITY

Once diagnostics itself is working reliably, it effectively becomes a testing tool. Diagnostics makes it easy to access internal variables and states in a standardized way – which is exactly what it was designed for. In this way, measured variables determined by the ECU can be used for control purposes in test benches, errors can be determined in road tests and reported to the engineers so they can be rectified, and the correct installation of components can be continuously checked in manufacturing.

The test sequences used must in turn be validated; errors in the test setup would make the entire methodology absurd. The entire test setup consisting of hardware and software has to be tested. And for many testers this is something that by no means happens just the once. As a rule, new vehicles or ECU variants are integrated on an ongoing basis. The actual state must be subjected to a regression test in each case to ensure that changes do not have any effect. To illustrate the point, take a service tester that is rolled out to 10,000 repair shops: Any errors are naturally going to lead to a new rollout and thus to enormous costs. This has to be avoided.

In both cases, experience points to a dilemma: To test a tester you need a remote station; in diagnostics a vehicle or an ECU. But normally these are not available. While a tester or test sequence is being created, they are not finished, which is hardly surprising as it first has to be verified by a test. In the regression test, they are usually no longer available because it is simply impossible to keep all vehicles in all variants.

And we have a similar picture in the training facility: The vehicle suitable for the subject matter of the exercise cannot be kept in stock and can only be procured at great expense.

## SIMULATION HELPS

In this case, the solution to the problem is obvious: a simulation that represents the ECU or vehicle in terms of diagnostic behavior. This should be able to realize real communication to be able to validate the entire chain: from the test sequence through the Vehicle Communication Interface (VCI) to the cabling. This ensures that when the Device under Test (DuT) arrives, in other words the vehicle or an ECU, any irregularities that occur can be clearly assigned to it.

This kind of simulation makes it possible, first of all, to simulate the communication and then, in a second step, the behavior, **FIGURE 1**.

Diagnostic communication usually follows the principle "request – response", in other words the tester sends a request for information to an ECU and the ECU sends the required information back. A response can be segmented, that means broken down into several submessages on the bus. If the ECU cannot supply the answer, it returns a negative response. Some information can only be accessed when an ECU is in a special state, for example when an engineer/developer session with its own authentication has taken place. A simulation must remember such states. All these communication mechanisms must be mastered to ensure diagnostic simulation is useful. In addition, it should also be possible to manipulate transmission contents, for example to be able to map changing values when information is requested more than once.

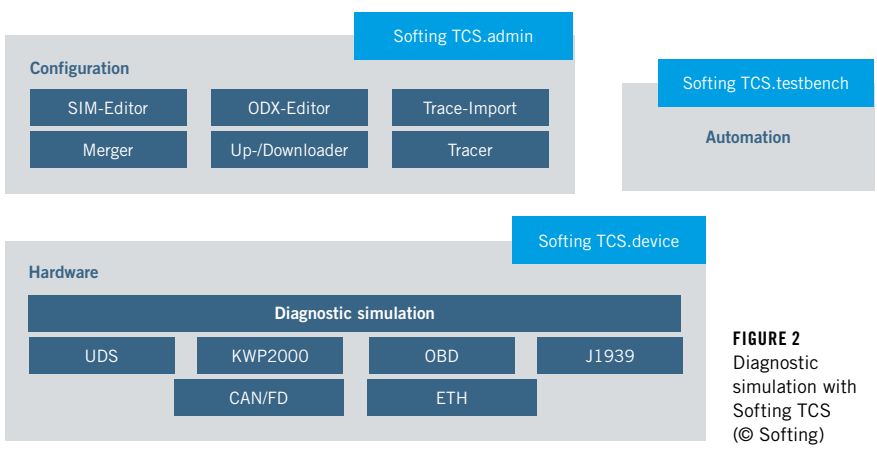



**FIGURE 2** Diagnostic simulation with Softing TCS (© Softing)

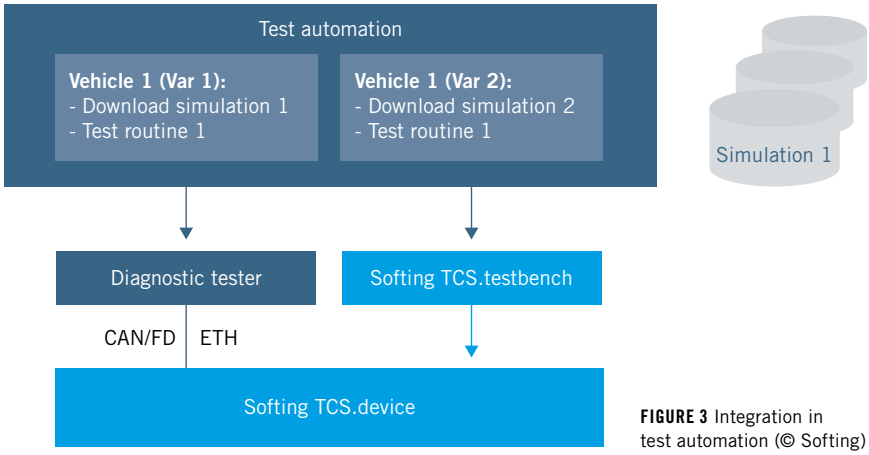



**FIGURE 3** Integration in test automation (© Softing)

In most cases, ECUs in different vehicles do not behave in the same way in terms of diagnostics. In practice, even different variants of an ECU in the same series differ. To make a simulation manageable, it has proven useful in practice to control the simulation with configuration files. A configuration created once for an ECU of a particular series is then easy to duplicate and adapt by making minor adjustments. In addition, test case developers can easily make adjustments to the simulation themselves.

## SOFTING TCS AS AN EXAMPLE

Softing TCS is an example of this kind of diagnostic simulation. The solution consists of three components TCS. device, to represents the hardware for the simulation, TCS.admin as the configuration interface for simulation files and the TCS.testbench to enable integration into test automation, **FIGURE 2**. The hardware, based on a Linux board with a multi-core processor, offers all interfaces to the vehcle that are relevant for the simulation case: In addition to CAN and CAN FD, communication can take place via Ethernet. An On Board Diagnosis (OBD) jack is available for flexible installation in test equipment. Softing TCS.device then simulates a vehicle. In addition, further sockets are available to cover development and a voltage supply can be connected. The integrated simulation unit communicates with the tester using standardized protocols such as Diagnostics over IP (DoIP), Unified Diagnostic Services (UDS) or OBD. Configuration takes place with Softing's own XIM format.

The XIM simulations are generated via the configuration interface Softing TCS.admin and loaded onto the hardware via the LAN. The simulation files are created in several ways, adapted to suit the different use cases. The specification of the ECU diagnosis is the basis in test preparation. Today, this is usually available as an ODX container. The simulation generator with an integrated ODX parser can generate simulations in two ways: first of all in Automatic mode, where simulation entries are generated for selected diagnostic services based on selected rules. Furthermore, in Expert mode, the corresponding response can be created precisely for each desired parameterization of a diagnostic service. The simulation data is represented symbolically and at message level. Incorrect responses can also be generated to simulate behavior with such responses.

Regression testing is covered by the generation of simulation files using trace records. For this purpose, the communication of available vehicles with the diagnostic tester is saved and then converted into a simulation. This takes place fully automatically. For this purpose, accepted tester requests are assigned the appropriate responses for each ECU. In addition, all settings can be checked and modified using a simulation editor. Special cases such as multiple responses and the editing of data content are also possible here, for example to test the coding in entirety.

All editing possibilities can be combined. For example, a simulation can be created via trace recording, then a new service can be added via the ODX specification, and behavior that deviates from the specification can be edited manually. This makes it extremely simple to create variants. Softing TCS.testbench provides a programming interface for integration into a test automation. This allows the exchange of simulation files and the targeted starting and stopping of a simulation, **FIGURE 3**.

## SUMMARY

Diagnostic simulation is an effective tool when there is no ECU or vehicle available: in test preparation, in regression tests and in training facilities. The targeted use of a solution such as Softing TCS makes it possible to significantly avoid mistakes. Testing can be more specific and in a wider range thanks to the time gained by earlier maturity. This means that the simulation not only saves a considerable amount of money, but also provides a basis for improving quality.

FOLLOW US:

**IMPRINT**
Special Edition 2022 in cooperation with Softing Automotive Electronics GmbH,
Richard-Reitzner-Allee 6/2. Stock, 85540 Haar; Springer Fachmedien Wiesbaden GmbH, Postfach 1546,
65173 Wiesbaden, Amtsgericht Wiesbaden, HRB 9754, USt-IdNr. DE81148419

**MANAGING DIRECTORS:**
Stefanie Burgmaier I Andreas Funk I Joachim Krieger

**PROJECT MANAGEMENT:** Anja Trabusch

**COVER PHOTO:** © Softing Automotive Electronics GmbH

optimize!
softing

# DIAGNOSTIC AND TEST SOLUTIONS BY SOFTING

AUTOMOTIVE
automotive.softing.com