for the first time. Figure 2 shows that the simulated emissions are indeed very close to those of the reference vehicle. The result is a considerable increase in the number deployment options for the HiL simulator in calibration work.

As a typical feature of the work packages related to diagnostics calibration, a large number of cycled exhaust gas measurements are required. The priority here is the definition of start-up conditions, the prioritization of exhaust gas emissions resulting from system faults and/or active diagnostic interventions. By shifting a segment of the required measurements to the HiL simulator, it can be used to perform an initial calibration, while functional cross connections and worst-case scenarios are determined early along the timeline. As a result, actual measurements taken on the test vehicle can be either dispensed with or used more effectively. Figure 3 shows the findings of a study investigating the comparability of the actual vs. simulated behavior of diagnostic functions in the European Driving Cycle. The comparison looks at the point in time at which the cycle flag is set, i.e., at which a diagnostic result is available. It was found that the results of the virtual vehicle were well within the range of the scatter of the real-world vehicle. For the lambda probe offset diagnostics, no scatter data was available for the real vehicle. The relatively large time difference was the reason for taking a closer look at the way the driver influences the results. It was found that a variation of the vehicle’s road speed (within the legally permitted tolerance of ± 2 km/h) may cause a shift in the result of well over 100 seconds. In contrast to the physical test vehicle, valuable robustness investigations of this kind can be run on the HiL simulator quite easily and cost-efficiently. And that is just another enthusiastic vote for HiL deployment.

Outlook In order to expand the benefits resulting from the deployment of simulators in ECU software development, it will be necessary to extend the existing model library and to continue the development of the requisite parameterization methods. To increase user acceptance of the model-based approach and win additional efficiency benefits, the configuration effort for application-specific models must be further reduced. In the future, the model-based approach can be more firmly anchored in the entire software development process, and be deployed much earlier along the development timeline. This includes also the shared utilization of models and simulation environments in function development and calibration.

THE CHALLENGE
To ensure competitiveness, the increased complexity and high diversity of variants must be mastered along with competitive pricing and high quality.

THE SOLUTION
It was demonstrated that the deployment of extended simulation models enables HiL systems to be used as a standard development tool also for calibration purposes. To this end, a virtual vehicle for LABCAR was developed.

THE BENEFIT
The results obtained with the virtual vehicle are within the scatter range of its physical counterpart. The result is a significant increase of the deployment options for the HiL simulator in calibration work. As a result, actual measurements taken on the test vehicle can be either dispensed with or used more effectively.

**Table 1:** Comparison of diagnostic function sequences in the physical vs. virtual vehicle.

<table>
<thead>
<tr>
<th>Diagnostic Function</th>
<th>Physical Vehicle</th>
<th>Virtual Vehicle</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catalyst diagnostics</td>
<td>860.5 s</td>
<td>862.8 s</td>
<td>+2.3 s / 0.3 %</td>
</tr>
<tr>
<td>Lambda probe offset diagnostics</td>
<td>895.1 s</td>
<td>966.2 s</td>
<td>+71.2 s / 8.0 %</td>
</tr>
<tr>
<td>Lambda probe dynamic diagnostics</td>
<td>272.3 s</td>
<td>316.5 s</td>
<td>-44.2 s / 12.6 %</td>
</tr>
<tr>
<td>Lambda probe dynamics diagnostics</td>
<td>65.8 s</td>
<td>71.1 s</td>
<td>-5.3 s / 8.0 %</td>
</tr>
</tbody>
</table>

Example: Instant of setting cycle flag in NEDC (New European Driving Cycle)
To control costs and at the same time increase quality, the auto industry has been promoting the standardization of diagnostic interfaces. Current established standards for the Application Programming Interfaces (APIs) between hardware and software are SAE J2534 (“Pass-Thru API”), ISO 22900-2 (“D-PDU API”), and TMC RP1210B. Adherence to these standards not only makes it possible for a PC-based diagnostic software to communicate with vehicle electronics with the aid of Vehicle Communication Interfaces (VCI) offered by a variety of manufacturers – it also replaces proprietary interfaces between hardware and software and eliminates manufacturer-specific protocols for communicating with the ECU.

In addition, the deployment of standard APIs is prescribed by legislation:

- The emission standards EURO V (passenger car) and EURO VI EPA3 (commercial vehicles) obligate vehicle manufacturers to ensure access to vehicle electronics for all repair facilities outside their network of service shops. A brief look at the EU Directive 642012 concerning EURO VI reveals how this is to be accomplished: “… A reprogramming of ECUs must be done in compliance with either ISO 22900-2, SAE J2534, or TMC RP1210B, and with the use of non-proprietary…”

As a provision of its Engineering Services, ETAS implements testing solutions for regression testing of diagnostic components in the ECU software or in diagnostic interfaces. These solutions subsequently support customers in their efforts to verify conformance with prescribed standards.

Automated regression testing

For a number of years, ETAS has performed the “classic” automation for testing the user interface and the lower level software layers with established tools, e.g., by means of Silk Tests or tests based on Visual Studio. Within the framework of API implementation on VCI platforms, ETAS has developed special regression tests for automated testing of the VCI-APIs. Jointly with customers in the U.S. and Europe, ETAS Application Engineering Services has extended these tests to run comprehensive integration tests and benchmarks. In its current form, the solution offers the following options:

- Fully automatic regression testing of VCI-APIs
- Fully automated testing and acceptance of implementations of standardized protocols in the ECU
- Checking for correct response
- Verification of specific return values, manufacturer-specific testing

The bottom line

As the customer’s service provider, ETAS ensures that the relevant procedures, sequences, and calls are reflected in test scripts. Because the test scripts are based on standard formats, suppliers of ECU software or diagnostic APIs can use them to identify and troubleshoot a given fault in their own environment. The exchange of sensitive data or of complex diagnostic software is not required. The acceptance is supported by detailed documentation. The OEM is thus in a position to outsource regression tests in their entirety to ETAS or third parties, while suppliers of ECUs and VCs are able to increase their quality by means of thorough pre-testing.

Figure 2: The test report provides developers and users with detailed troubleshooting information. Testing environment and test scripts ensure test traceability and can be submitted to suppliers.

Figure 3: UDS Protocol Testing Demonstration.

- Flexible extension, adaptation, and exchange of test cases by means of xml-based test scripts
- Testing of various hardware units, with optional conformance check of the respective VCI-API
- Extensibility for proprietary interfaces (e.g., metrology or legacy hardware)
- Use and configuration of existing simulation environment (HiL, SiL, ECU, third-party simulator)

Conformity testing of Pass-Thru VCs

In one application of the testing environment, a customer checks Pass-Thru VCs for standards conformance. Regardless of a positive or negative test result, this involves the testing of all API calls of relevance to the customer on actual vehicles or on pre-configured simulators.

UDS conformity testing

In another use case, and as part of the acceptance of ECUs and their software updates, the testing environment is used to validate the support of those UDS protocol services (Unified Diagnostic Services, to ISO 14229) that are actually used. This is accomplished by executing on the ECU all of the services in use in conjunction with all relevant combinations of possible protocol parameters:

- Testing for positive & negative ECU responses
- Checking for correct response lengths
- Timing behavior testing

Figure 1: The selection of the appropriate testing environment is based on the type and degree of test automation to be performed.