ATZ extra

Springer Vieweg

SOVD

SDE

Diagnostic Systems
Accelerate the Development

optimize!
softing

Diagnostic Systems Accelerate the Development

The repair of today's vehicles with their network of 50 bis 150 computers, several bus systems and distributed functions is no longer possible without an expert system based on the control unit diagnosis. The same applies to production: errors cannot be found without a tester. In order for the diagnosis to work, considerable effort is required in vehicle development – and the trend is increasing, as Softing Automotive describes.

The diagnostic basis is still the same as it was ten years ago: The diagnostic protocol used is UDS (Unified Diagnostic Services); the diagnostic description is made in ODX (Open Diagnostic Data Exchange). When these standards were first specified, there were considerably fewer ECUs installed in vehicles. Two to three vehicle buses were all you needed and the amounts of data to be programmed were infinitesimally smaller.

In recent years, ASAM e.V. specified a new diagnostic standard to meet the new requirements: Service Oriented Vehicle Diagnostics (SOVD) [1]. The API (Application Programming Interface) was defined for a runtime system implemented on the vehicle. This makes diagnostic information available for applications which either run directly in the vehicle, for example a programming application for over-the-air updates, or can be accessed externally. For this purpose, the API essentially follows the Representational State Transfer(REST) paradigm. So, in principle, it can be operated via Hypertext transfer protocol secure (https). The "Service Oriented" in SOVD is a reference to the fact that entire blocks of information are read out as opposed to individual pieces of information, which is the common procedure today. This also makes it easy to set up remote applications because the quality of the transmission path is irrelevant for performing diagnostics [2].

Basically, the standard is perfectly suitable for use in manufacturing and after-sales service. For this purpose, it requires a suitable control unit with the ability to exchange data with the environment, such as a Telematics Control Unit (TCU). If the implementation is correspondingly mature, it can also prove very useful in engineering. There are a few limiting conditions to be considered here.

USING DIAGNOSIS IN ENGINEERING

To put it simply, diagnostic engineering must be divided into two phases, that is the engineering of the mechatronic system and the Integration.

In the engineering of the mechatronic system, the communication protocol first has to work in terms of diagnostics. Based on this, the actual diagnostic functionality is created in the control unit:
– monitoring routines which lead to error memory entries
– access to measurement values
– parameterization possibilities of the diagnostic software, for example for variant coding
– programming possibilities for the software update.

These functions are then verified within the ECU test with a simulated environment like Hardware in the Loop (HiL) and on the test bench together with the real environment.

As a result, several control units are first integrated into buses and their communication and functionality are tested in interaction. Then, vehicle prototypes can be set up and ultimately diagnostic functions released in the real vehicle during the road test.

As a result, several control units are first integrated into buses and their communication and functionality are tested

WRITTEN BY

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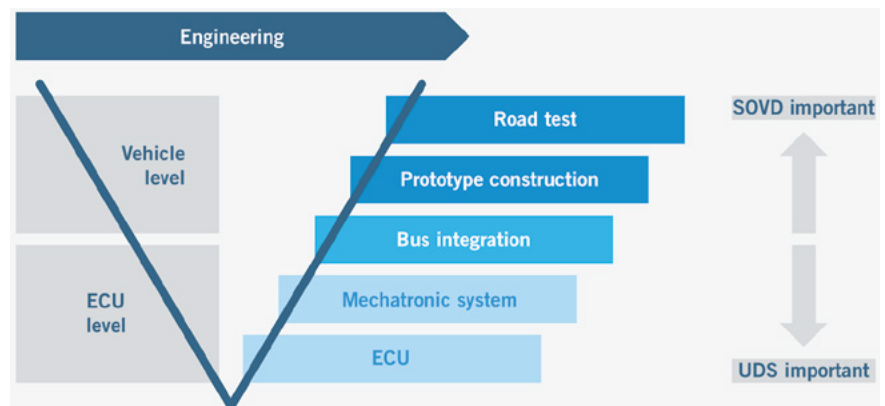
© Softing Automotive

FIGURE 1 Changing diagnostic requirements in development (© Softing Automotive)

in interaction. Then, vehicle prototypes can be set up and ultimately diagnostic functions released in the real vehicle during the road test, **FIGURE 1**.

From the point of view of the diagnostic application, this results in two completely different scenarios. The SOVD server is usually available in the assembled vehicle. Diagnostics should therefore also take place via this server to be able to ensure sufficient maturity for manufacture and for the application in service. The same applies to bus integration in some cases, but this depends on the E/E architecture and the position of the SOVD server within that architecture. During ECU engineering, however, diagnostics continues to be performed via UDS; after all, the SOVD server is not yet available at this point, or not available to every ECU engineer. This poses several challenges for the architecture of the diagnostic system.

IMPACT ON
THE DIAGNOSTIC SYSTEM

As a consequence, diagnostic systems must support two different paths: Initially, diagnostics takes place over UDS, but with increasing maturity and greater integration of the whole system, over the SOVD server. Parameterization is necessary in both systems. Engine ECU diagnostics quite obviously differs from a door ECU. This is taken into account in current diagnostic tools through the use of data ODX, which describes the different diagnostic capabilities. The situation with the SOVD server is similar as it prescinds the diagnostics of the whole vehicle which is equipped differently. For example, a station wagon has a tailgate control unit, whereas a sedan doesn't. This has to be made accessible with parameterization and must also be able to be changed during operation, as numerous modern functions are implemented in software and can thus be adapted after sale, **FIGURE 2**.

In general, a diagnostic system thus has one path, via which an ODX system enables diagnostics with ECUs via UDS, and a second path, via which an SOVD server that has to be parameterized using proprietary methods is the diagnostic basis. Both paths can certainly also be used in parallel. For

example, an ECU or function manager will need to safeguard their area of responsibility via the SOVD server, after all, this is how diagnostics will be used in productive operation in manufacturing and after-sales. If problems occur, direct access to the ECU via the UDS path makes it much easier to pinpoint the cause.

IDEALIZED SOLUTION APPROACH

In such a hybrid system, the SOVD server can be parameterized in the same way as the MVCI server: via the above-mentioned ODX data. In real systems, these are rarely used in their pure form, but in a runtime format. This has the advantage that the data can be protected more easily as a safety-critical component and the runtime data can be both highly compressed and runtime optimized. In addition, the hybrid system shows consistent runtime behavior across both paths, making the results equally comparable and reliable.

If you can also transfer the SOVD part of the hybrid system to the vehicle, you have a uniform system for both ECU diagnostics and vehicle diagnostics. This makes the results equally reliable across the entire value chain. For in-vehicle use, the data will simply be reduced to the necessary extent. ODX data (Open Diagnostic Data Exchange) typically contains all possible vehicle variants in order to enable diagnostics – for example in the repair shop environment – with every possible installation or software variant. Both are known in the vehicle which means that only the correct data needs to be kept available in each case.

So the target image might look like this:
– There is a runtime system that is used in the vehicle in a SOVD server.
– The runtime system can work with vehicle specific and complete data.
– For ECU diagnostics it is operated as an MVCI server on the PC.
– For vehicle diagnostics in engineering, it is operated as an SOVD server on the PC.

The result would mean everything: uniform runtime behavior across all use cases, optimized data sizes in each case – and a validated diagnostic system in the vehicle.
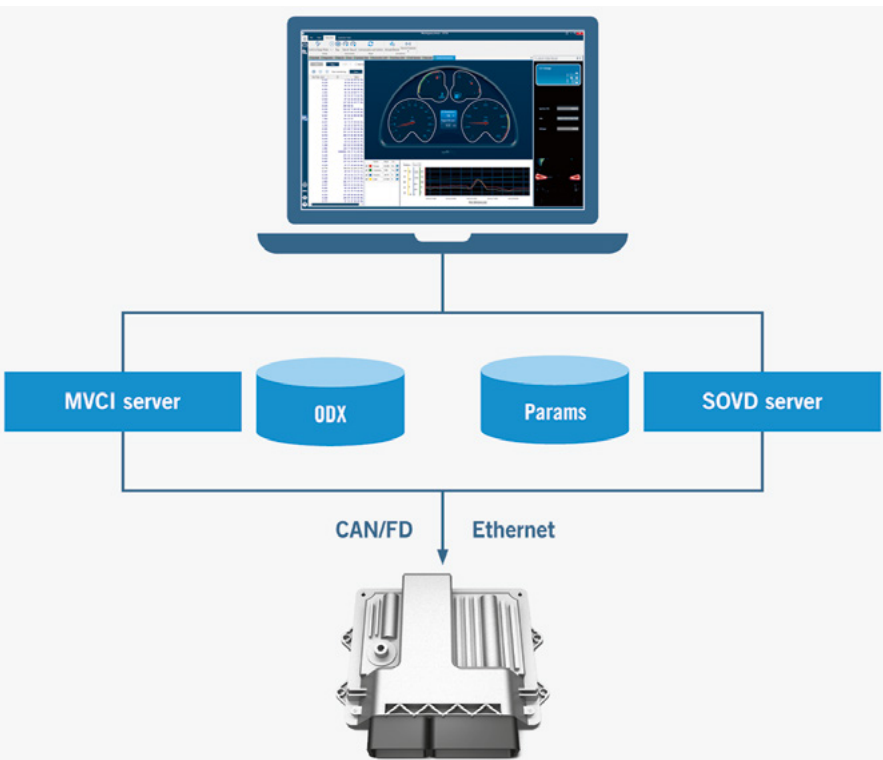
FIGURE 2 In the future, vehicle functions will be implemented primarily in software, so it must also be possible to change the parameters during operation (© Softing Automotive)

DIAGNOSTIC RUNTIME ENVIRONMENT
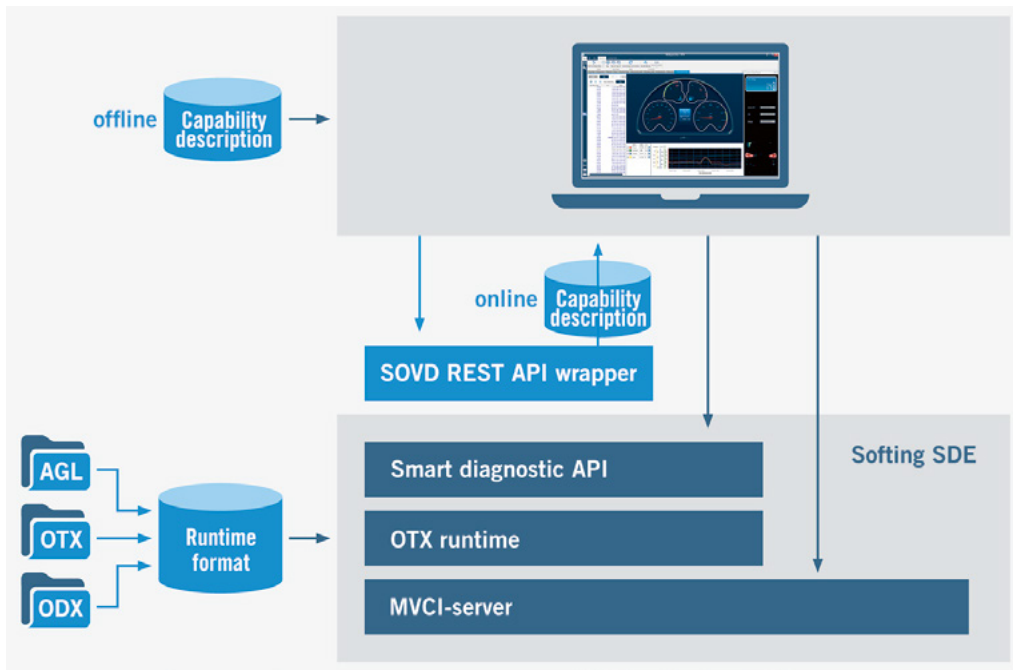
Softing SDE is an ODX-based, platform-independent diagnostic runtime environment which has been available for many years. It is used in numerous applications in engineering, manufacturing and after-sales. The MVCI server is combined with an ODX runtime environment (Open Diagnostic Data Exchange) for standardized diagnostic sequences in compliance with ISO 13209. The Smart Diagnostic API also provides a service-oriented interface that offers applications entire diagnostic functions.

One example is reading the error memory, which combines several UDS services (reading errors for each ECU, querying environmental conditions for each error) into one function call on the SDE. This is easy to learn and use in remote scenarios. The functions then run independently of the con-
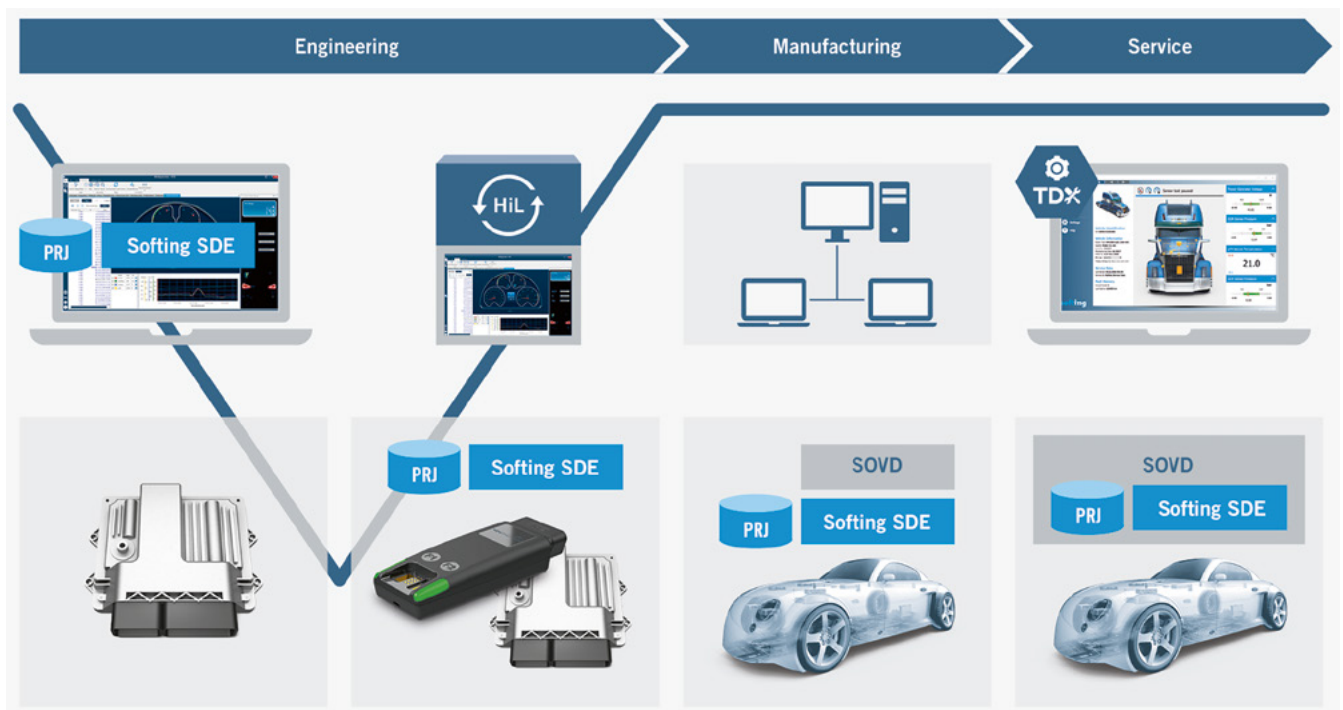
FIGURE 4 SOVD as REST interface for the SDE (© Softing Automotive)

nection link like 4G/5G: Results can basically be retrieved when the connection is sufficiently good, **FIGURE 3** and **FIGURE 4**.

These service-oriented functions correspond in the abstraction level to the functions required by SOVD. This makes it very easy to fulfill the standard: Only the existing C++ API has to be converted via a wrapper to a REST interface in accordance with the SOVD standard. Depending on the application, the Capability Description described in the SOVD standard (this contains the diagnostic capabilities of a vehicle like the ODX data), can then be generated online from the ODX data if necessary.

The procedure enables a continuous diagnostic chain from ECU engineering and the test environment to manufacturing and into the repair shop.First, diagnostics is implemented locally in the ECU and verified using the engineering tester. The integrated SDE then continues to be used in test benches with exactly the same data and configurations. Localization separate from the test sequence is easily possible, for example in a VCI. In the road test, the SOVD implementation based on the SDE is then used and thus the final methodology released. In turn, in manufacturing you can only work

with the SDE as long as the ECU with the SOVD server is not installed or cannot yet be used. In the finished vehicle, diagnostics is then only carried out via the SOVD server, **FIGURE 5**.

Remote scenarios are also possible at any time with this procedure [3]. Test benches can be operated remotely at any time via the SDE and in the road test, this can be taken care of both using proprietary methods and with the SOVD server. This is equally possible on the customer vehicle – after all, Remote is a main use case for SOVD.

CONSISTENT DIAGNOSIS

SOVD changes diagnostics, the standard offers great advantages not least in the interaction of several partners via the Internet. However, additional methodologies are necessary in the early engineering phases, and the standard simply has not been developed for ECU diagnostics. Here, particular attention must be paid to data processes, which can lead to significant additional costs in parallel systems. Diagnostic systems such as Softing SDE, which have been designed for a wide range of application scenarios and can be easily extended with an SOVD API, make it possible to leverage all the advantages.

REFERENCES
[1] Bernd Wenzel ASAM e.V.; Tobias Weidemann Vector Informatik: SOVD – Service oriented Vehicle Diagnostics. Munich, October 07./08. 2021, online. chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.asam.net/index.php?eID=dumpFile&t=f&f=4485&token=aae-0a0499792b569d82fd860d07d1b5b0de049d9, access: June 20th 2023
[2] Steffelbauer, M.: SOVD - The Diagnostic Standard of Tomorrow. In: ATZ Electronics 16 (2021), Nr. 4, S. 44-48
[3] Steffelbauer, M.: Biggest Possible Development Efficiency with Remote Engineering. In: ATZelectronics 14 (2019) Nr. 10, S 36-39

FOLLOW US ON SOCIAL MEDIA:

DIAGNOSTIC AND
TEST SOLUTIONS
BY SOFTING

AUTOMOTIVE
automotive.softing.com