Originally published in the section Engineering Tools of the German technical magazine HANSER automotive, issue 12/2012

SOFTING: DIAGNOSTIC TOOL SET VERSION 8

Diagnostics: Communication Server for all Standards

Using a standardized runtime system for diagnostic communication immensely simplifies and accelerates the development of user applications. In this article, we present a new communication server that supports MCD-3D 3.0, ODX 2.2 and DoIP as well as drastically reduces tester memory requirements thanks to its ultra-compact runtime format.

In today's vehicles, the number of ECUs and the associated networking are continually increasing. In addition to the actual control functions, diagnostic functions have also gradually become more and more significant. In the past, vehicle manufacturers invested incredible amounts of both time and money to develop their own, proprietary systems for ECU communication. These work with non-compatible formats for data description and necessitate the use of special vehicle interfaces in each case. When no appropriate standards are available, costs are immense and manufacturers can become dependent on specific suppliers. This is why vehicle manufacturers, system suppliers and tool manufacturers got together to specify and implement a whole range of international standards. Alongside the communication protocol UDS, a standardized diagnostic runtime system and its interfaces are of particular importance for diagnostics. Since a communication server is a major component, it is also often referred to as a D-Server or MVCI-Server. Its main task is to receive, process and send diagnostic data. During processing, the machine-readable representation becomes readily under-standable for the user. This runtime system is used throughout the entire value chain:

- In Engineering & Testing: release of ECUs, creation and validation of test sequences, HiL systems
- In Manufacturing: end-of-line test systems, test benches, programming stations
- In After-Sales Service: basic software for service testers

DTS 8 supports these standards

- **ISO 22901-1/ASAM MCD-2D V2.2.0 (ODX)**
- **ISO 22900-2 (D-PDU API) via CAN and K-line**
- **ISO 22900-3/ASAM MCD-3D V3.0.0**
- **ISO 13400 - Diagnostics over Internet Protocol**

Internationally standardized interfaces

The ISO 22900-1 (ASAM MCD-2D/ODX) standard defines a data model and universal exchange format for the comprehensive description of diagnostic communication and programming data. This includes protocols, communication parameters, vehicle interfaces including connectors and their pin assignment, flash data as well as ECU variants. ODX has created a completely new way of working: For the very first time, the engineering, testing, manufacturing and after-sales service departments as well as all suppliers can use the same database for diagnostic tasks.

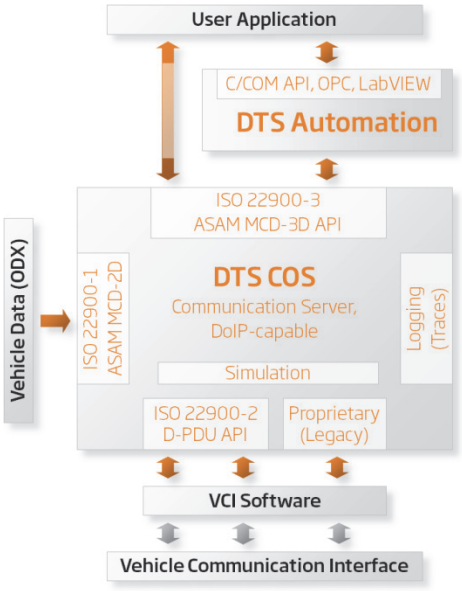

Figure 1: System Overview of Diagnostic Runtime System DTS COS and DTS Automation

The exchange of diagnostic specifications can take place in accordance with the single-source principle over a vehicle's entire lifecycle. OEMs usually compile all data for a particular model range in one database. ODX databases of current vehicles can be relatively extensive. A special binary runtime format is thus generated to increase performance: It contains only the required data without redundancies and optimized for access.

The bus system interface in accordance with ISO 22900-2 makes it possible to use all kinds of bus protocols via interfaces from different manufacturers with standardized "drivers" (D-PDU API). From the point of view of the application, the bus protocol remains entirely transparent. The implementation of a D-PDU API is a one-off effort for a VCI interface. The description of the capabilities of the relevant interface is stored in an XML file. This makes it extremely easy to change the interface. For example, a test program can be operated with a USB interface made by manufacturer A in engineering and then later with a WLAN interface made by manufacturer B on the test bench.

The application interface in accordance with ISO 22900-3 (ASAM MCD-3D) allows symbolic access to ECU and vehicle information without detailed knowledge of the bus protocols used being necessary. This takes place using the conversion methods, service and data type descriptions contained in the ODX database. For example, an "Engine" ECU can be addressed by name. To determine the engine speed, the "ReadSpeed" service specified for exactly this purpose is used. The ECU response is then used to calculate the physical variable with value and unit as 1900 min^{-1}.

Fully compatible with the standards

The diagnostic runtime system DTS COS is part of Softing's Diagnostic Tool Set product family that makes it possible for developers, engineers and technicians to create consistent diagnostic functions and sequences on the basis of international standards and to ensure that diagnostic communication works reliably over the entire value chain. DTS COS implements all above-mentioned standards in entirety – see Figure 1. Complete adherence and real-time performance are verified by extensive tests in accordance with the official ASAM test suite. With the communication server and the standardized interfaces, DTS COS enables users to focus on the actual applications. Using a special "simulated" interface, test sequences and user applications can even be tested without real ECUs. Extensive trace functions enable developers and engineers to quickly detect errors in their own applications or in the communication with the ECUs. Trace files can be analyzed "offline" with the Analyzer. Using a special ConfigAPI, the runtime system can be configured entirely by an external application in terms of interfaces, projects etc. The optional API Developer Kit facilitates the development of applications based on the DTS communication server. In addition to extensive documentation and programming examples, it also contains a special TestApp. This makes it possible for developers to establish communication to the vehicle via the runtime system immediately, i.e. without their own application development.

Benefits the new version brings the User

The latest international standards (see box on previous page) are supported in Version 8, available at the end of 2012. In ODX 2.2 and particularly in ASAM MCD-3D 3.0, there is less room for interpretation which means OEM authoring guidelines on diagnostic data descriptions will converge which in turn will reduce dependencies on a particular tool manufacturer. Existing ODX 2.0.1 data, now widely implemented globally, can easily be reused if necessary without changes having to be made. Thanks to an optimized runtime format, vehicle data, flash data and jobs become anything from 60 to 180 times more compact in comparison to the ODX database depending on the OEM and ODX structure – see Figure 2. This means less RAM is required in the tester and updates are quicker to run. A considerable benefit as diagnostic data for one model range nowadays often amounts to more than 100 MB.

A major German OEM is acting as a pilot customer and is already successfully using the new, OEM-specific runtime data encryption. This represents an incredible increase in security both in terms of data exchange between OEM and suppliers and later in the case of unauthorized access to a service tester in the repair shop. With the completely overhauled System Configurator, it is now even easier and more intuitive to manage system and project settings centrally. Assistants are available to help import data and create new projects. The new Database Differ makes it possible to compare databases directly in the runtime formats. The Differ is a convenient way of finding changes made between the different vehicle statuses.

Simple Access

For the realization of test benches in ECU and vehicle manufacturing, simplified access to the ODX database is required via the standard interfaces widely used in industrial automation. Based on DTS COS, DTS Automation offers an API reduced to the scope of this application. The API could thus be considerably simplified because of this. The communication mechanisms it is based on are transparent for the user. DTS Automation can either be accessed via a C-DLL, OPC or LabVIEW VIs.

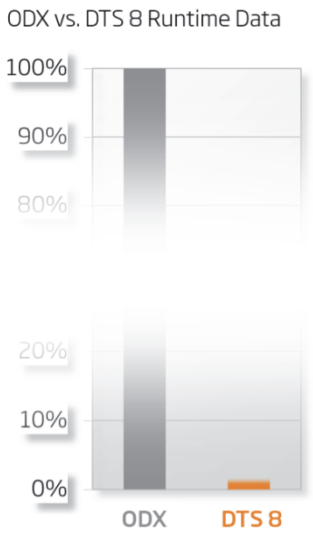

Figure 2: Ultra-compact runtime data

```
        ..
'--- Init and Startup ------------------------------------------
Set API = New dtsPcAPI       'Create API object
Set Sys = API.dtsInit("DTSDemoConfiguration.cfg", 1)
                             'Init DTS using Automation CFG file
Set AccessPath = Sys.loadAccessPath("DiagCan_ECU1")
                             'Set database location
Set MV_Temperatur = AccessPath.getMeasurementValue("Temperatur")
                             'Create value "Temperatur"
'--- Function body ----------------------------------------------
MV_Temperatur.ECUstartRead  'Read "Temperatur" from ECU
Temperatur = MV_Temperatur.ECUreadValueLong
        ..
```

Figure 3: This is how easy it is to access a temperature via the DTS Automation API.

Figure 3 shows an example of how simple it is for a user application to read out a temperature measure value. Depending on the configuration, objects or variables that can be combined to form a test sequence in the test bench application are available at the interfaces. Every test sequence can be saved in a special configuration file thus enabling very short changeover times.

Summary

Using a standardized runtime system for diagnostic communication considerably simplifies and accelerates the development of proprietary applications. Communication and programming data can be specified consistently in ODX. Engineering, testing, manufacturing, after-sales service and all suppliers can exchange diagnostic specifications over a vehicle's entire lifecycle in accordance with the single-source principle. The D-PDU API bus system interface allows both the use of a whole range of bus protocols via interfaces made by various manufacturers and a simple change between different interfaces. The MCD-3D 3.0 application interface allows symbolic access to ECU and vehicle information without detailed knowledge of the bus protocols used being a prerequisite.

The diagnostic runtime system DTS COS implements the latest international standards in entirety. Powerful tools make it possible for developers to test, for example, test sequences and user applications without real ECUs and to quickly detect errors in proprietary applications or in the communication with the ECUs. In addition to ODX 2.2.0, the new version supports the unproblematic reuse of existing ODX 2.0.1 data and offers an ultra-compact, OEM-specific encryptable runtime format.

DTS Automation gives test bench applications particularly easy access to vehicle diagnostics via automation interfaces.

The products presented enable users to focus on the actual applications and get results fast.

Matthias Ziegel is Product Manager for the Diagnostic Tool Set at Softing Automotive Electronics GmbH in Haar, Germany.

Andreas Vach is Manager Vehicle Communication and as such heads up the development of the runtime system for the Diagnostic Tool Set at Softing Automotive Electronics GmbH in Haar, Germany.