Originally published in the German technical magazine Automobil Elektronik, issue 5/2013

Optimized Diagnostics

Implementing Offboard Diagnostics More Efficiently with OTX

In the past, the consistent creation of test sequences in vehicle diagnostics was not a reliable process due to the implementation of a whole range of tool and language concepts. Test sequences were specified as free text and implemented on all kinds of target systems.
With the OTX (Open Testsequence eXchange) standard, the automotive industry has for the very first time created a language format with which diagnostic sequences can be specified and programmed without having to change tools. The XML-based format makes it possible to exchange validated sequences regardless of departmental and process boundaries and to store them securely long term.

Author: Matthias Meyer, Softing Automotive GmbH

Today, sequence information tends to be created and exchanged in freely phrased and freely formatted documents which are processed with all kinds of different tools. An ECU expert may well start off by specifying the test sequences in Visio, Word or Excel. Once the specification has been completed, it is printed out and handed on to a specialist for tester programming. This specialist then converts the test sequences to the relevant language of his or her preferred automa-

tion tool. It is pretty obvious that this kind of process is prone to error and is thus relatively expensive.

A further disadvantage is that test sequences created in this way cannot always be reused in entirety. Due to the large number of languages and tools currently in circulation, there is no uniform, platform- and technology-independent format for the documentation and execution of test sequences. And it is therefore virtually impossible to ensure that test routines, once created, can still be used reliably in years to come.

Fig. 1: The world before OTX: use of all kinds of formats for test sequence creation

OTX – The Standard

OTX supports users both in the specification of processes and in subsequent implementation. As a specification language, OTX enables the graphic creation of sequence plans with freely worded description texts. As a programming language, this specification is converted into individual concrete, executable actions with OTX. The OTX specification allows for procedure calls, actions, branches and loops, as well as elements for fault diagnosis and, last but not least, processing mechanisms for the simultaneous execution of sequences.

Although OTX features a lot of characteristics of a programming language, it is not a classic programming language with its own syntax. Because OTX uses a standardized archiving format for both the specification and the implementation – with the major advantage that processing on a PC is thus considerably easier to realize. Like other standard exchange formats, OTX uses XML for this purpose. An OTX run-time component reads in the sequences in XML format and then takes care of working through them.

In addition to the program structuring elements already mentioned, the standard defines a range of function libraries for specific application cases in a separate part: These considerably extend the performance scope of the OTX sequence language. It is only in this part of the standard that vehicle diagnostics comes into play. The central OTX data model ("OTX core") is independent of this as OTX was developed for use in a number of different environments.

The following are just a few of the most important standard libraries included with OTX: In addition to accessing vehicle diagnostics (including a library for ECU flashing), OTX contains tools for graphic user interaction (HMI Library), a library with mathematical functions, an add-on for dealing with multiple languages (Internationalization, "i18n") and possibilities for converting physical values with measuring units (Quantities).

Integration into Existing Vehicle Diagnostics Standards

OTX is very closely related to the existing standards ISO 22900 (MVCI) and ISO 22901 (ODX). The following figure shows how OTX is integrated in these standards. As has already been explained, this does not mean that OTX can only be used in this context. But it does become apparent that the most important area of use of OTX is vehicle diagnostics to close the gap mentioned at the beginning that previously existed in terms of the reliable description of test sequences.

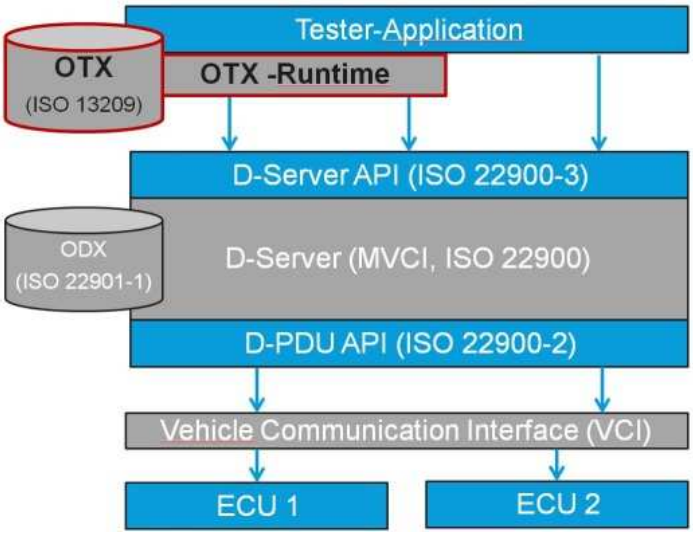

Fig. 2:Integration of OTX into existing vehicle diagnostics standards

The standard with the similar-sounding name, ODX ("Open Diagnostic data eXchange", ISO 22901), describes the diagnostic-relevant services and data of an ECU and for this purpose also uses an exchange format in XML. The MVCI Standard (ISO 22900) describes a diagnostic system with standardized interfaces to the calling application (D-Server API) and to protocol software (D-PDU API).

If an ODX description is loaded onto the D-Server, a tester application can access it via the D-Server API. The user selects the required diagnostic service which in turn is sent to the ECU as a communication telegram.

OTX uses communication via the D-Server API in a similar way for the connection to the vehicle. The standardized vehicle diagnostic libraries are a convenient way of accessing this interface. The link is via the OTX runtime which calls the D-Server and controls the diagnostic sequence process.

Specifying and Programming Sequences in One Format

OTX supports a flowing development process: At a very early stage, known as the "specification stage", future test sequences can be created in an initial form even though not all sequence details are known. They are specified as free text, already stored in the target format XML and can be exchanged.

Once the specification stage has been completed, an appropriate editor is used to fill the free-text sequences with instructions which can be processed by the runtime system (OTX runtime). The script can already be run in this partially implemented state ("intermediate stage"). Once the intermediate stage has been completed, all test steps defined in the specification stage have been extended with executable instructions. The completed script is now implemented in entirety and can be run in full ("realization stage").

As diagnostic experts can create their own sequences, the effort involved in specification is greatly reduced. The time previously needed for consulting the

experts on how to convert sequences into an executable format has thus been considerably shortened and at times is not actually necessary at all. Instead of the time-consuming, complex creation of automated diagnostic sequences in programming languages, something that necessarily involved a high degree of specialist knowledge, OTX scripts can now be created faster and at a much reduced cost with a very short familiarization time.

OTX makes reviews and continued development of test sequences much faster and simpler than before. Changes to different version releases of the test sequences are easy to follow and document. Furthermore, a specification view in the form of a flow chart is the perfect tool for coordinating the opinions of all those involved in creating and modifying OTX scripts.

Due to the high availability of OTX sequences, even minimal room for improvement can be used to advantage in practice as scripts are easy to modify and fast to put back into productive use. All in all, this flexibility represents significant cost savings as any optimization potential detected is quick to put into practice.

Another important point is that specification and implementation are united in one single document. This means that the documentation of specialist knowledge collected in sequence creation during the specification stage becomes an integral part of the development process in the creation of diagnostic sequences. The OTX sequence is thus documented in a readable fashion for further use without any extra effort.

OTX Studio

Appropriate tool support is required to make optimal use of the advantages of the OTX standard. Softing's "OTX Studio" is a development environment with an extensive function scope. OTX sequences that have been created are compiled together with all the relevant data as projects. Debug function such as breakpoints, single-step processing and variable monitoring ensure user-friendly working during error location.

Different views can be consulted depending on the area of use. Within these views, the user can store frequently used standard sequences as individual libraries. ISO conformity of the created scripts is tested using an integrated OTX-Checker; this enables unproblematic exchange of OTX sequences.

Conclusion

The use of OTX means test sequences can be exchanged without having to change tools between the different links of the value chain. Although not exclusively designed for vehicle diagnostics, OTX contains all elements necessary in the field of diagnostic communication for the consistent specification and implementation of sequences.

The user is thus capable of generating productive test applications and diagnostic sequences fast. Consistent use thereof results in considerable cost savings. Uniform creation, long-term re-use and the exchangeability of validated sequences are thus ensured.

Matthias Meyer
Product Manager
at Softing Automotive
Electronics GmbH in Haar,
Germany.