



Diagnoseentwicklung mit AUTOSAR

Der AUTOSAR Diagnostic Extract eröffnet neue Möglichkeiten in der Diagnoseentwicklung. Von der präzisen Beschreibung der Daten inklusive der Ableitung der Diagnose-BSW-Konfiguration, über die verteilte Entwicklung der Diagnose beim OEM und Tier1 bis hin zu einem Top-Down-Ansatz zur automatischen Integration der Diagnosefunktionen im Steuergerät. Die Toolkette von Vector nutzt diese Möglichkeiten und stellt gleichzeitig die Synchronität der ODX- und DEXT-Daten sicher.



© Vector Informatik



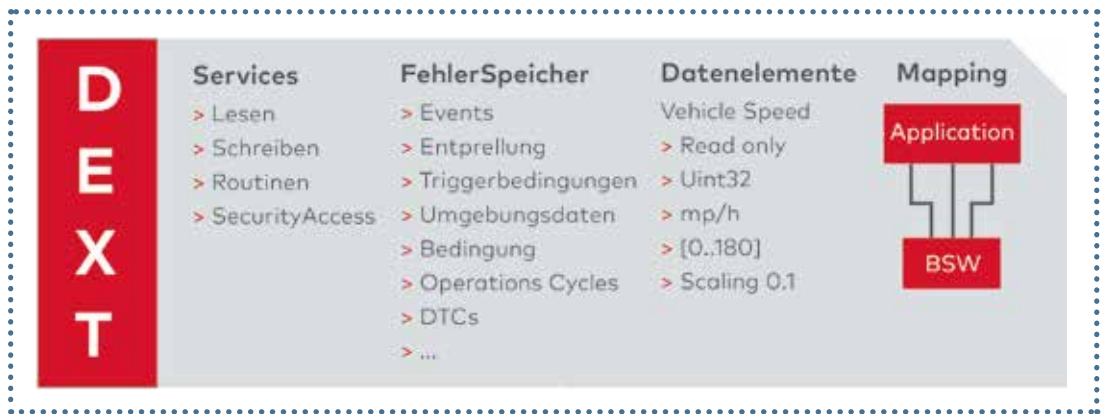
Den Entwicklungsprozess für die Fahrzeugdiagnose gestalten OEMs und Tier-1-Zulieferer in unterschiedlichster Form. Es kommen verschiedene Austauschformate zum Einsatz und die eingesetzten Werkzeuge unterstützen meist nur den jeweils eigenen Prozess. Probleme treten spätestens dann auf, wenn Diagnosebeschreibungen mit Entwicklungspartnern ausgetauscht und in deren Toolkette weiterverwendet werden sollen. Dabei kommt es immer zu erhöhtem zeitlichem Aufwand, falls es denn überhaupt gelingt, Daten ohne Informationsverlust auszutauschen. Mit dem AUTOSAR Diagnostic Extract Template (DEXT) bieten sich ganz neue Möglichkeiten für die Diagnoseentwicklung. Der diagnoserelevante Teil der Basissoftware kann einheitlich über Unternehmensgrenzen, beispielsweise OEM und Tier1 oder auch in OEM-Kooperationen, hinweg konfiguriert werden. Aufgaben, die bisher dem Integrator beim Tier1 überlassen waren, können vorweg dezentral über DEXT konfiguriert werden.

DEXT im Vergleich zu anderen Diagnosedatenformaten

Veröffentlicht worden ist DEXT erstmalig mit AUTOSAR 4.2.1. Zunächst wurde die Beschreibung der mittels UDS transportierten Daten und der UDS-Fehlerspeicher standardisiert. Mit AUTOSAR 4.3.0 kamen entsprechende Erweiterungen für OBD-II, WWH-OBD, FIM (AUTOSAR Function Inhibition Manager) und SAEJ1939 hinzu. Damit deckt DEXT inhaltlich die in AUTOSAR unterstützten Basissoftware-Module für die Diagnose ab. Mit DEXT lassen sich nicht nur die über das jeweilige Protokoll transportierten Daten beschreiben, sondern auch die Herkunft der Daten in der Applikationssoftware des Steuergeräts. Nur wenn beide Informationen vorliegen, kann die Diagnose-Basissoftware vollständig konfiguriert werden. Das Diagnoseprotokoll, insbesondere die Beschreibung der Diagnosedienste, und die Übertragung der Daten auf dem Netzwerk werden nicht beschrieben. Hier



Bild 1: Elemente des AUTOSAR Diagnostic Extract.
(© Vector Informatik)



stützt sich AUTOSAR auf die Vorgaben des UDS- und OBD-II-Standards.

Für die Diagnosetester wiederum hat sich ODX als Datenformat für generische Tester durchgesetzt. ODX beschreibt das Diagnoseprotokoll sowie die zwischen Steuergerät und Tester transportierten Daten mit deren Interpretation in einem Diagnosetester. Die Quelle der Daten im Steuergerät spielt für die Verarbeitung der Daten im Tester keine Rolle und fehlt daher in ODX. Trotzdem kommt ODX für eine initiale Bedatung in Frage, die in erster Linie die Existenz und die Struktur der Diagnosedaten auf der Leitung beschreibt. Die Verknüpfung der Diagnosedaten mit der Steuergeräteapplikation muss dann allerdings manuell oder durch spezielle Prozessschritte erfolgen. In der Fehlerpeicherbeschreibung ist die Informationslücke zwischen ODX und Steuergerätesoftware am größten. So sind beispielsweise die verwendeten Entprell- oder Verdrängungsalgorithmen in der Fehlererkennung elementar wichtig für die Basissoftware, in ODX fehlt diese Information aber. Die teilweise sehr unterschiedlichen Autorenrichtlinien der Fahrzeughersteller erschweren die Austauschbarkeit der Steuergerätekonfigurationen zusätzlich.

Bedatung der Basissoftware

Prozesse, die das AUTOSAR-ECUC-Format für Datenaustausch und Vorbedatung der Basissoftware verwenden, füh-

ren in der Praxis selten ans Ziel. Das ECUC-Format ändert sich häufig und wird mit jeder neuen Version des Standards angepasst. Zudem wurde es in erster Linie als Eingangsformat für die Embedded-Software-Codegeneratoren konzipiert. ECUC ist zusätzlich offen für herstellerspezifische Erweiterungen, was es als neutrales Austauschformat untauglich macht. AUTOSAR DEXT ist auf die Anforderungen, welche sich für die Bedatung der Basissoftware ergeben, zugeschnitten. Die wesentlichen Bestandteile sind (Bild 1):

- Auswahl der Diagnosedienste und zugehöriger Subservices für UDS, OBD, WWH-OBD und J1939,
- Fehlerpfad und Fehlerpeicher,
- Diagnose-Datenelemente und deren Paketierung,
- Mapping der Diagnose-Datenelemente in die Applikationssoftware,
- Querverriegelung (FiM).

Das folgende Beispiel zeigt anhand eines Data-Identifiers (DID) die Vorteile des AUTOSAR Diagnose Extracts. Wie ein DID zu modellieren ist, wird durch das AUTOSAR-Metamodell formal festgelegt. Anders als bei ODX gibt es hier keinen Interpretationsspielraum und damit auch keine Missverständnisse im Datenaustausch zwischen Tools. Die Existenz eines DIDs wird durch die Instanz eines „DiagnosticData-Identifiers“ angegeben. Diese Instanz enthält die für UDS notwendige 16-bit-Nummer des DIDs. Außerdem aggregiert diese Instanz ein oder mehrere Datenelemente, wodurch die »

Szenario 1 Vollständiges Design der Diagnose durch den OEM	Szenario 2 TIER1 erweitert die OEM-Diagnosespezifikation	Szenario 2 TIER1 erstellt die Diagnosespezifikation
<ul style="list-style-type: none"> > OEM erstellt Systemdesign (incl. SWCs mit Diagnoseschnittstellen) > OEM stellt die ECU Diagnosespezifikation zur Verfügung > TIER1 übernimmt die ECU Diagnosespezifikation vom OEM 	<ul style="list-style-type: none"> > OEM erstellt Systemdesign (incl. SWCs mit Diagnoseschnittstellen) > OEM stellt die ECU Diagnosespezifikation zur Verfügung > TIER1 erweitert das ECU Diagnosedesign > OEM aktualisiert seine Diagnosespezifikation mit den Modifikationen des TIER1 	<ul style="list-style-type: none"> > TIER1 erstellt SWCs mit Diagnoseschnittstellen > TIER1 erstellt die Diagnosespezifikation passend zu den SWCs > OEM übernimmt TIER1 Diagnosespezifikation in seine Spezifikation

Tabelle 1: Typische Diagnoseprozesse zwischen OEM und Tier 1. (© Vector Informatik)

Daten des DIDs mit Name und Typ definiert werden. Bei der Datentypmodellierung hat AUTOSAR das bereits vorhandene System Template Metamodell wiederverwendet. Der DID selbst kann von den Serviceinstanzen „DiagnosticReadDataByIdentifizier“, „DiagnosticWriteDataByIdentifizier“ oder „DiagnosticIOControl“ durch Referenz auf den DiagnosticDataIdentifizier in einen Diagnoseservice verwendet werden. Damit ist die Definition eines DIDs abgeschlossen und kann als Quelle für die Bedatung der Diagnose-BSW (Basissoftware) dienen.

Diagnostic-Mapping

Fehlt noch die Interaktion der BSW mit der Applikationssoftware zum Lesen, Schreiben oder Übersteuern des DIDs. Hierfür gibt es im DEXT ein weiteres Element: das Diagnostic-Mapping. Es beschreibt die Verbindungen zwischen den Diagnoseelementen in der Basissoftware, wie Routinen, Daten eines DIDs, Events, und den SWCs (Software Components) aus der Applikationsschicht. Die SWCs müssen hierfür eine geeignete Schnittstellenmodellierung aufweisen, die einem von AUTOSAR definierten Modellierungsmuster folgt. Es gibt verschiedene solcher Muster für den Zugriff über einen Funktionsaufruf mittels Client/Server-Schnittstelle oder aber als Datenkommunikation über eine Sender-/Receiver-Schnittstelle. Früher musste der Integrator teilweise tausende dieser Verbindungen zwischen den Ports der Basissoftware und der Applikationssoftware manuell konfigurieren. Mit dem Einsatz von DEXT kann diese Tätigkeit automatisiert beim OEM/Tier1 erfolgen. Der Integrator übernimmt das Mapping anstatt es manuell für viele Verbindungen selbst zu erzeugen. Die Fehleranfälligkeit sinkt bei gleichzeitiger Zeitersparnis und Qualitätssteigerung.

Szenarien und Rollen einer verteilten Diagnoseentwicklung

Heute unterscheiden sich die Prozesse in der Diagnoseentwicklung in der Praxis erheblich. Neben den eingesetzten Tools und Austauschformaten in den verschiedenen Tools, gibt es große Unterschiede in dem Beitrag von OEM und Tier 1. In der Praxis findet man alle möglichen Spielarten, vom vollständigem Design durch den OEM, über eine Kombination aus Diagnoseentwicklung zwischen OEM und Tier 1 bis hin zur vollständigen Entwicklung beim Tier 1. Tabelle 1 gibt einen Überblick über typische Diagnoseprozesse.

Basierend auf DEXT lässt sich jede der drei Prozessausprägungen unterstützen. Wie in allen AUTOSAR arxml-Formaten sind auch bei DEXT fast alle Elemente optional. In den einzelnen Prozessschritten kann DEXT entweder initial er-

stellt, angereichert oder final ergänzt werden. Die erstellten Daten sind in sich immer valide und können ausgetauscht werden. Welche Daten dabei in welchem Prozessschritt hinzugefügt werden ist unerheblich und wird nur durch den verwendeten Prozess definiert.

Anforderungen an eine Toolkette zur Unterstützung dieser Prozesse

Eine Toolkette im Diagnoseentwicklungsprozess muss an die oben genannten Szenarien anpassbar sein. Am Anfang des Prozesses steht das Festlegen der Diagnoseanforderungen im Requirements Management System (RMS). Aus einer Diagnoseanforderung ergeben sich wiederum Anforderungen an die Applikationssoftware sowie Anforderungen an die dazu passenden Diagnoseservices. Beide Anforderungen werden in der Regel von verschiedenen Rollen getrennt umgesetzt – in Form von SWCs sowie in Form einer geeigneten BSW-Konfiguration. Zum Zeitpunkt der Steuergeräteintegration werden sie durch den Integrator zusammengeführt. Ge-

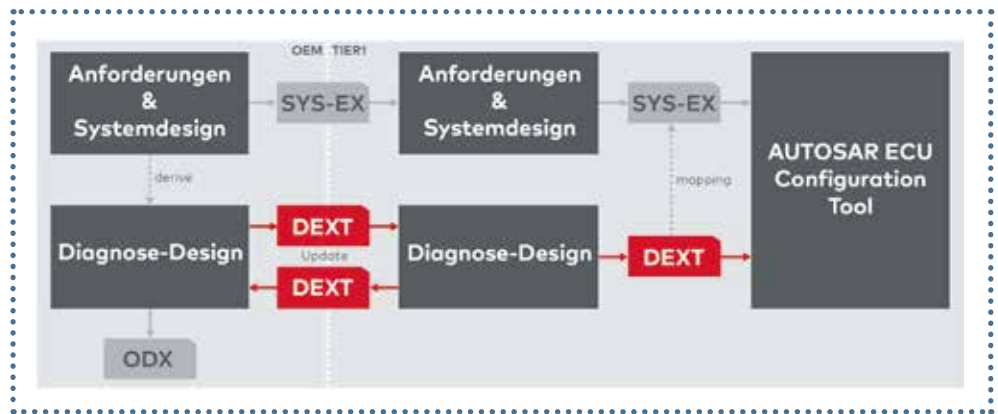


Bild 2: Diagnoseprozess zwischen OEM und Tier 1. (© Vector Informatik)

nau hier kommt das erwähnte Diagnostic-Mapping in DEXT zum Einsatz. Der gesamte Prozess ist in Bild 2 aufgezeigt.

In einem Top-down-Prozess wird die Diagnose-Applikationssoftware zuerst entwickelt beziehungsweise eine bestehende Software wiederverwendet. Aus den Anforderungen und Schnittstellenbeschreibungen der Applikationssoftware leitet sich die Diagnosebedatung ab. Darin besteht ein wesentlicher Vorteil im Vergleich zu vielen existierenden Prozessen, in denen die Diagnosedaten in oft sehr aufwendiger manueller Abstimmung an die Anforderungen und die Applikationssoftware angepasst werden.

Parallel zur Erstellung des DEXT-Formats müssen auch ODX-Daten entstehen. Generiert man ODX- und DEXT-Daten aus einer gemeinsamen Quelle, kann sichergestellt werden, dass die Diagnosesoftware im Steuergerät und der Diagnostiktester zueinander passen.

Beispiel anhand einer Toolkette

Eine Toolkette zur Diagnoseentwicklung die diese Anforderung abdeckt, zeigt Bild 3 mit Werkzeugen von Vector beispielhaft. Sie besteht aus den Tools PREEvision für Anforderung

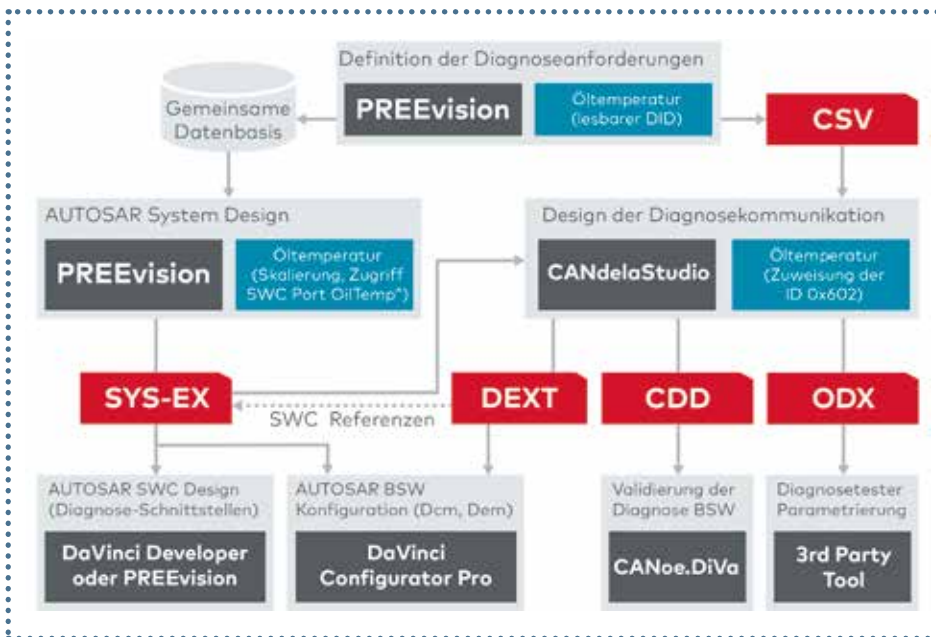


Bild 3: Diagnoseworkflow beispielhaft dargestellt anhand einer Toolkette von Vector.

(© Vector Informatik)

rungen und Systemdesign, CANdelaStudio als Diagnose-Autorentool und DaVinci Configurator Pro zur Konfiguration der BSW. In PREEvision wird zu einem frühen Zeitpunkt in der Entwicklung eine Anforderung an die Diagnose definiert. Dies könnte zum Beispiel die Anforderung an einen Öltemperatursensor sein. Für diesen Sensor soll es in der Diagnose einen Service zum Lesen der Öltemperatur geben, einen Service für das Überschreiben des Sensorwertes über IO-Control sowie ein oder mehrere mögliche DTCs, die ein Fehlerverhalten des Temperatursensors anzeigen.

Aus diesen Anforderungen werden die SWC-Schnittstellen des System-Extracts im arxml-Format abgeleitet. Die Schnittstellen der SWC geben dabei auch die Parameter der Diagnoseobjekte vor. Im Beispiel des Öltemperatursensors liefert ein Port der SWC den aktuellen Temperaturwert, und die Schnittstelle des Ports definiert, ob es sich um einen 16- oder 32-Bit-Messwert handelt, welche Umrechnungsformel und welche Einheit verwendet wird. Eine spezielle für diesen Workflow entwickelte SWC-Sync-Funktionalität in CANdelaStudio erstellt daraus automatisch die passenden Diagnose-daten für folgende Diagnoseelemente:

- Diagnostic Data Identifier (DIDs) für Read, Write und IO-Control,
- Routine Control Identifier (RIDs),
- Events und DTCs.

In unserem Beispiel legt der Diagnoseexperte in CANdelaStudio einen „ReadDataByIdentifier“-Diagnoseservice mit einem Datenelement „Temperatur“ als vorzeichenloser 16-bit-Wert in 0.1-Kelvin-Auflösung an, ein IO-Control Service mit dem gleichen Datenelement und einen DTC, der einen defekten Sensor signalisiert. Er legt auch den Identifier fest, über den die Öltemperatur per Diagnose erreichbar ist. Gleichzeitig merkt sich CANdelaStudio, welcher Port an der Softwarekomponente die Temperatur liefert und welcher Diagnoseservice von diesem Port die Daten liest. CANdelaStu-

dio kann damit das DEXT-Format inklusive des Diagnostic-Mappings exportieren. DaVinci Configurator Pro liest das DEXT-Format ein und leitet die Konfiguration der Diagnose-BSW-Module daraus ab. Anschließend erzeugt DaVinci Configurator Pro die SWC-Schnittstelle der Diagnose-BSW-Module und verbindet sie mit den Ports der Applikations-SWCs – passend zu den Diagnose-Mappings im AUTOSAR DEXT.

Zusammenfassung und Ausblick

Der AUTOSAR Diagnostic Extract eröffnet neue Möglichkeiten in der Diagnoseentwicklung. Von der präzisen Beschreibung der Daten inklusive der Ableitung der

Diagnose-BSW-Konfiguration, über die verteilte Entwicklung der Diagnose beim OEM und TIER1 bis hin zu einem Top-Down-Ansatz zur automatischen Integration der Diagnosefunktionen im Steuergerät. Die Toolkette von Vector nutzt diese Möglichkeiten und stellt gleichzeitig die Synchronität der ODX- und DEXT-Daten sicher.

Anwendung findet AUTOSAR DEXT in beiden AUTOSAR-Plattformen: initial für AUTOSAR Classic entwickelt, ist es auch in AUTOSAR Adaptive das einzige Beschreibungsformat der Diagnose. Die vorgestellte Methode lässt sich also auch für Entwicklungen auf Basis der AUTOSAR-Adaptive-Plattform einsetzen. ■ (oe)

» www.vector.com

» www.hanser-automotive.de/6683130

Hier finden Sie die Download-Version des Beitrags.



Dipl.-Inf. Wigbert Knappe ist Produktmanager in der Produktlinie Embedded Software und Systeme bei Vector in Stuttgart.



Dipl.-Ing. (FH) Matthias Wernicke ist Produktmanager AUTOSAR Tools in der Produktlinie Embedded Software und Systeme bei Vector in Stuttgart.



Dr. Klaus Beiter leitet ein Entwicklungsteam in der Produktlinie Diagnose bei Vector in Stuttgart. Er ist Mitglied in der ASAM/ISO-ODX Arbeitsgruppe.