

Status Quo AUTOSAR

Der Automobilindustrie stehen in den nächsten Jahren große Herausforderungen bevor. Die Fahrzeugelektronik hat bereits heute eine kaum noch zu beherrschende Komplexität erreicht. Die Kunden der OEMs wünschen jedoch eine noch höhere Varianten- und Ausstattungsvielfalt. Außerdem haben die Automobilhersteller Kriterien zu beachten, die für den Endkunden in der Regel nicht erlebbar sind. Zu diesen nicht-funktionalen Anforderungen zählen beispielsweise Diagnosefähigkeit, Verfügbarkeit oder Upgrade-Fähigkeit. Für Zulieferer stellt sich die Lage ebenfalls kompliziert dar. Sie haben nicht nur die Modellvielfalt eines OEMs zu beherrschen, sondern müssen diese Anforderungen für mehrere Kunden erfüllen. Hinzu kommt, dass sich der Zulieferer auch an den unterschiedlichen Produktentstehungsprozessen bei seinen Kunden orientieren muss. Dies betrifft den Entwicklungsprozess, die Elektronikarchitektur und nicht zuletzt auch die Tool-Landschaft der Unternehmen.

Mit der Elektronik-Initiative AUTOSAR will man der Problematik unterschiedlicher, größtenteils proprietärer Lösungen entgegenwirken und somit den Entwicklungsaufwand reduzieren. AUTOSAR unterteilt dazu die Elektronikarchitektur in mehrere Schichten und Module, definiert deren Schnittstellen und schafft so Standards für einen einfachen Austausch von Softwarekomponenten oder Hardwareplattformen. Darüber hinaus wird durch AUTOSAR ein Designprozess vorgegeben, der mit einer modellbasierten Funktionsbeschreibung beginnt, durchgängig Tool-basiert weitergeführt wird und im Idealfall in einem automatisch generierten und reproduzierbaren Test endet.

Das Release 1.0, seit Mai 2005 verfügbar, spezifiziert 31 Basissoftware-Module (BSW). Die anschließende Implementierungs- und Validierungsphase (Validator 1) bestätigte die Funktionsfähigkeit der Module und des Konzepts. Im Mai 2006 konn-

Standardisierte Architekturen und Schnittstellen sollen komplexe Systeme beherrschbar machen

Die ungeheure Vielfalt an Betriebssystemen, Embedded Software, Mikrocontrollern und Softwarekomponenten von verschiedenen Herstellern verursacht einen erheblichen Integrationsaufwand bei verteilten Elektronikarchitekturen. Diesem Dilemma hat sich die AUTOSAR-Entwicklungspartnerschaft mit der Definition von Standards angenommen. Der Beitrag beleuchtet den aktuellen Stand der AUTOSAR-konformen Steuergerätesoftware-Entwicklung.

Von Matthias Wernicke und Jochen Rein

ten bereits 42 von 46 BSW-Spezifikationen veröffentlicht werden, darunter auch die Runtime Environment (RTE). 99 Dokumente wurden bislang fertiggestellt, von denen 94 veröffentlicht sind. Derzeit durchlaufen die Spezifikationen eine weitere Testphase (Validator 2), die im Herbst endet. Das erste vollständige Release (R 2.1), das dann auch die Methodikspezifikation beinhaltet, ist für Ende 2006 geplant.

Modularer Aufbau bietet Vorteile für alle Partner

Hard- und Software eines Steuergeräts waren in der Vergangenheit immer untrennbar miteinander verbunden. Steuergeräte mussten bei zusätzlichen Merkmalen, neuen Controllern, anderen OEMs oder geänderten Architekturen meist von Grund auf neu entwickelt werden.

Das Konzept der AUTOSAR-Initiative baut auf modularen Komponenten auf. Steuergeräte-Zulieferer können dadurch beispielsweise Applikationskomponenten entwickeln, die sich in Projekten verschiedener OEMs wiederverwenden lassen. OEMs portieren Funktionen zukünftig problemlos über ihre Fahrzeugplattformen hinweg, ohne notwendigerweise stets das gleiche Steuergerät für die Ausführung einer bestimmten Funktion wählen zu müssen. Zudem erleichtert die strukturierte Herangehensweise eine Modularisierung der Applikationssoftware,

die Bereitstellung von zuverlässiger, wieder verwendbarer Standardsoftware und ein einfacheres Software-Update über den gesamten Lebenszyklus eines Fahrzeugs. Dadurch können sich alle Partner auf ihre Kernkompetenz konzentrieren und sind von der ständigen Neuentwicklung von Basisfunktionen weitgehend entlastet.

AUTOSAR vereint Hardware, Basis- und Anwendungssoftware

In AUTOSAR wurden alle Bestandteile eines Steuergeräts abstrahiert und in Anwendungssoftware, Basissoftware und Hardware-Ebene unterteilt (Bild 1). Die unterste Schicht stellt die Hardware-Ebene dar. In ihr werden alle Hardware-Features des Mikrocontrollers abstrahiert. Fehlende Hardware-Features werden an dieser Stelle durch entsprechende Softwaremodule ega-

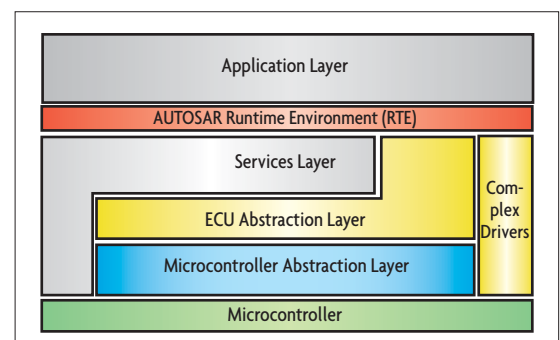


Bild 1. AUTOSAR-Schichtenmodell. Durch Module mit standardisierten Schnittstellen werden die Mikrocontroller- und ECU-Spezifika von der Anwendungsebene abstrahiert.

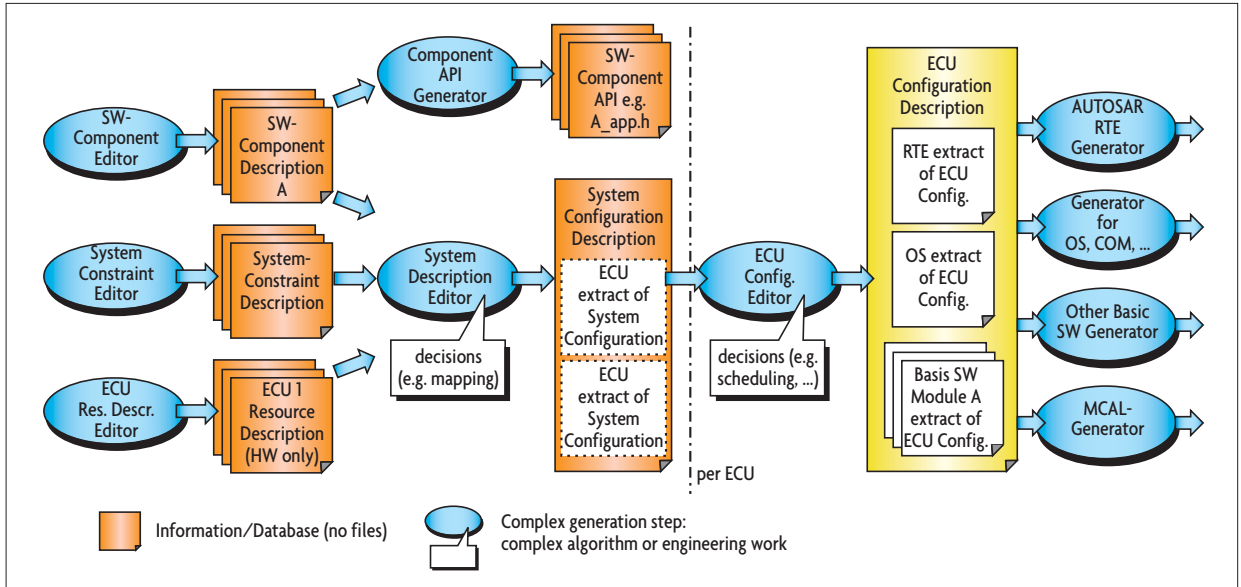


Bild 2. Strukturbeschreibung der Softwarekomponenten. Die Erstellung von AUTOSAR-konformen Softwarekomponenten ist in klar vorgegebene Entwicklungsschritte unterteilt. (Quelle: Vector Informatik GmbH)

lisiert. Diese SPAL (Standard Peripheral Abstraction Layer) genannte Schicht beinhaltet Mikrocontroller-Treiber, Speicheransteuerung, Kommunikations- und I/O-Treiber.

Über dem SPAL steht die Abstraktionsebene der ECU. Diese Schicht abstrahiert alle Basiskomponenten, die sich in der ECU befinden. Hier befinden sich auch die Treiber für externe Peripheriekomponenten. Die nächsthöhere Schicht, Services Layer, ist bereits weitgehend von der Hardware unabhängig und stellt verschiedene Arten von Hintergrunddiensten wie Speicherverwaltung, Netzwerk- oder Buskommunikationsdienste bereit. Der RTE als vierte Schicht kommt eine besonders wichtige Bedeutung zu: Sie verbindet die Komponenten der Anwendung mit der Basissoftware, indem sie den Datenaustausch und die Interaktion zwischen ihnen regelt.

Um diese Schichten konfigurieren zu können, wird das System zunächst in einer abstrahierten Sicht auf Basis des Virtual Functional Bus (VFB) entworfen. Diese Sicht beschreibt die Anwendungssoftware in Form von miteinander kommunizierenden Komponenten. Zwischen den Komponenten erfolgt die Kommunikation über Ports, deren Schnittstelle über das Port Interface definiert ist. Die Schnittstelle kann entweder für Datenkommunikation oder für Funktionsaufrufe gemäß dem Client/Server-Prinzip ausgeführt

sein. Konnektoren verbinden die Ports miteinander. Dabei spielt es auf Systemebene keine Rolle, ob die Konnektoren intern innerhalb eines Steuergerätes oder extern durch Kommunikation über einen Datenbus realisiert werden. Erst bei der konkreten Systemauslegung entscheidet sich dies durch die Zuweisung der Softwarekomponenten auf die einzelnen ECUs.

Die Softwarekomponente selbst hat kein Wissen über diese spätere Aufteilung und kann daher unabhängig vom Systemkontext entwickelt werden. Die Ausführungseinheiten einer Softwarekomponente werden Runnable Entities genannt. Sie entsprechen Prozeduren und werden bei Eintreten eines bestimmten Ereignisses ausgeführt, wie etwa einer zyklischen Aktivierung oder bei Ankunft eines neuen Eingangswertes.

Aus der gemäß VFB-Sicht erstellten Beschreibung des Systems ergibt sich die Programmierschnittstelle (API) der Softwarekomponenten, die über die RTE bereitgestellt wird. Mittels dieser API lassen sich Eingangswerte lesen oder Ausgangswerte schreiben. Auch für den Zugriff auf die Basisdienste, wie etwa dem Memory-Service, kann eine API bereitgestellt werden.

Durch das Prinzip der modellbasierten Beschreibung der Software-Architektur sind RTE und Basissoftware skalierbar und können ressourcenscho-

nend an die jeweiligen Anforderungen angepasst werden.

AUTOSAR definiert die Methodik der Software-Entwicklung

Neben den Vorgaben zur Bestimmung der Architektur definiert AUTOSAR auch die Methodik der Software-Erstellung. Hierin liegt ein wesentliches Element der Entwicklung zuverlässiger Komponenten: Eine strukturierte Herangehensweise zeigt früh fehlende oder falsche Anforderungen, vereinfacht die Wiederverwendung von Softwarekomponenten sowie deren Portierung und führt zu einem zuverlässigeren System. AUTOSAR setzt dabei auf werkzeuggestützte Entwicklungsprozesse. Die Methodik dient der strukturierten Herangehensweise, lässt aber dennoch genug Freiheiten für unterschiedliche Entwicklungsprozesse.

Dank Werkzeugunterstützung erstellen die Entwicklungsteams die System- und Steuergerätekonfigurationen systematisch und konsistent. Die komplexen Abhängigkeiten der Daten werden von den Werkzeugen automatisch berücksichtigt. Die drei Hauptbestandteile Software (SW Component), ECUs (ECU Ressource) und System-Randbedingungen (System Constraints) werden zunächst formal beschrieben. Durch geeignete Editoren entsteht daraus eine Konfigurationsbeschreibung des kompletten Systems. Diese System

Configuration dient als Basis zur Erstellung der Steuergeräte-Konfigurationen (ECU Configuration), die vom Anwender mithilfe geeigneter Konfigurationswerkzeuge für die einzelnen Basissoftwaremodule erstellt werden. Mehrere Generatoren liefern am Ende des Prozesses die Steuergeräte-spezifische Implementierung von RTE und Basissoftware.

Wichtig ist, dass alle während des Entwicklungsprozesses entstandenen Entwurfs- und Konfigurationsdaten in einem einheitlichen Format beschrieben werden können. Dazu wurde ein Format auf XML-Basis (Extensible Markup Language) definiert, das die Durchgängigkeit des Entwicklungsprozesses garantiert und das nahtlose Einfügen notwendiger oder zusätzlicher Tools erleichtert.

■ **Tool-Unterstützung für den Entwurf verteilter Funktionen**

Ohne leistungsstarke Autorenwerkzeuge ist eine Entwicklung von AUTOSAR-Komponenten undenkbar (**Bild 2**). Ein manuelles Erstellen der für den Entwurfsprozess benötigten XML-Dateien ist auf Grund der Komplexität nicht möglich. Vector Informatik hat mit seiner DaVinci Tool Suite bereits 2003 eine Werkzeugfamilie geschaffen, die genau den von AUTOSAR heute favorisierten strukturierten und modularen Ansatz verfolgt und die Entwicklerteams beim Entwurf verteilter Funktionen, bei der Zuordnung von Softwarekomponenten und bei der Code-Integration der vernetzten Steuergeräte unterstützt. Als einer der ersten Toolanbieter unterstützt der Premium-Partner der AUTOSAR-Initiative einen durchgängigen und konsistenten AUTOSAR-Entwicklungsprozess, mit dem sich alle Schritte systematisch abarbeiten und die komplexen Zusammenhänge kontrollieren lassen. Die Toolunterstützung reicht vom Entwurf der Software-Architektur und der Netzwerkkonfiguration bis zur Codegenerierung. Im Hintergrund

überwacht die Werkzeugumgebung eASEE das Daten- und Konfigurationsmanagement, die Benutzerverwaltung und die Datenintegration.

Zur Beschreibung des Gesamtsystems erstellt der Anwender mit dem DaVinci System Architect zunächst eine formale Beschreibung der Softwarekomponenten und der Systemtopologie aus ECUs und Netzwerken. Danach weist er die Softwarekomponenten den einzelnen ECUs zu und

legt damit fest, welche Kommunikation lokal und welche über ein Bussystem erfolgt. Die Portdaten werden den Bussignalen zugeordnet und die Signallaufwege definiert, z.B. über Gateway-Steuergeräte. Mittels DaVinci Network Designer werden das Layout und das Zeitverhalten (Scheduling) der Netzwerkbotschaften definiert. Ergebnis ist die System Configuration Description; diese enthält die vollständige Information über die Systemarchi-

Eine Möglichkeit ist der frühe Umstieg auf die RTE. Dadurch wird die Anwendung weitgehend unabhängig von den darunter liegenden Schichten. Allerdings verlagert sich der Aufwand in Richtung RTE. Diese muss die darunter liegenden Varianten unterstützen, sofern die Varianten nicht bereits ein AUTOSAR-konformes API zur RTE zur Verfügung stellen. Die Applikationsentwicklung mit einer RTE bietet sich für den Zulieferer bereits heute an. Er kann sich unabhängig vom Fahrzeughersteller dafür entscheiden und auf ein standardisiertes API aufbauen, noch bevor der OEM dies fordert. Damit ist er frühzeitig unabhängig von weiteren Migrationsschritten (Bild 4). Für die Konfigurationsdaten sind ebenfalls Migrationsschritte notwendig. So müssen heutige Standardformate wie DBC, LDF oder FIBEX in einheitliche AUTOSAR-XML-Formate überführt werden.

Die erste, im Frühjahr 2006 abgeschlossene AUTOSAR-Spezifikationsphase hat gezeigt, dass an den konzeptionellen Überlegungen kaum Veränderungen notwendig sind. Die derzeitige Validierungsphase 2 wird unter anderem Aufschluss geben, wie der Mehraufwand bei Speicher und Rechenlast der CPU im Steuergerät zu bewerten ist.

■ Entwicklungsumgebung für Ausführung und Simulation

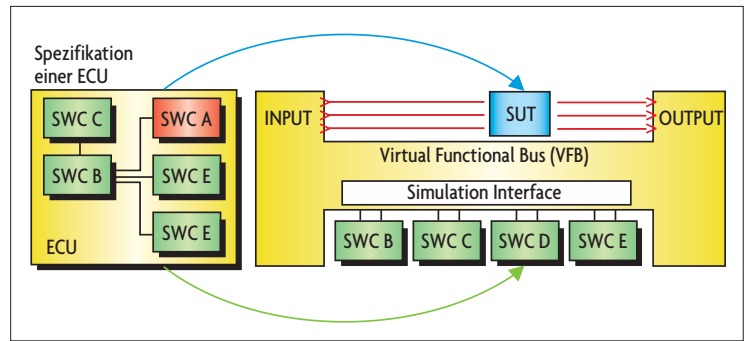
Da sich der Standard als umfassender Entwicklungsansatz versteht, ist die Verwendung einer integrierten Entwicklungsumgebung (IDE) wie Visual Studio oder Eclipse als Basis für den gesamten Prozess nur konsequent. Als AUTOSAR-Entwicklungsumgebung sollte sie idealerweise eine Ausführungsplattform für einzelne Softwarekomponenten und eine Simulationsplattform für das Gesamtsystem bieten, ergänzt durch eine Steuer-, Test- und Visualisierungsumgebung (Bild 5).

Vorteile ergeben sich insbesondere durch die Möglichkeit, frühzeitig Tests durchzuführen. Anwender können schon während der Entwicklung komfortabel Testzenarios erstellen und müssen nicht erst die vollständige Implementierung der Softwarekomponenten abwarten. Testfälle lassen sich

einzelnen abarbeiten und Software wird sowohl einzeln als auch integriert aus der IDE kontrolliert ausgeführt. Durch die Fähigkeit, dies langsamer oder schneller als in Realzeit auszuführen, ist ein effektives Debugging möglich; eine Vielzahl automatisierter Tests ist in kurzer Zeit durchführbar. Mit der Anbindung von CANoe an die IDE können die gleichen Testfälle mit Teilsystemen auch unter Einbeziehung realer Netzwerke ausgeführt werden.

Die Kernidee von AUTOSAR ist es, durch Standardisierung die wachsende Komplexität der Software in modernen Fahrzeugen zu beherrschen. Das Ziel, eine offene Referenzarchitektur für Steuergeräte-Software zu definieren, wird mit dem ersten vollständigen Release 2.1 Ende 2006 erreicht. Durch Standardisierung und eindeutig spezifizierte Schnittstellen zwischen den Basissoftwaremodulen und zur Applikationssoftware ergeben sich viele Möglichkeiten wie etwa die Wiederverwendbarkeit der gleichen Software für verschiedene Mikrocontroller und Applikationen bei gleichzeitig steigender Qualität.

Für alle bei der AUTOSAR-konformen Software-Entwicklung notwendige



! Bild 5. Test einer Softwarekomponente in der Ausführungs- und Simulationsplattform. Während eine ausführbare Softwarefunktion real getestet wird, können zusätzliche Funktionen einfach als Modelle über das Simulationsinterface ergänzt werden.

(Quelle: Vector Informatik GmbH)

gen Entwurfsschritte bietet Vector die entsprechende Lösung, vom strukturierten Entwurf der AUTOSAR-Softwarekomponenten und deren Verteilung auf Steuergeräte, über die Definition der Kommunikation bis hin zur Konfiguration der effizienten Basissoftware. Der Migrationsaspekt ist dabei ein wichtiger Schwerpunkt. So kann der Anwender vorhandene Entwurfsdaten in Formaten wie z.B. DBC oder FIBEX mit AUTOSAR-Entwurfsdaten kombinieren und existierende Basissoftware in AUTOSAR BSW überführen. Die offene Lösung begünstigt nicht nur die Integration weiterer Werkzeuge in die Toolkette, sondern ermöglicht auch eine konsequente Weiterentwicklung der Tools gemäß der aktuellen AUTOSAR-Spezifikationen, an denen Vector durch aktive Mitarbeit im Konsortium richtungweisend beteiligt ist. sj



**Dipl.-Ing. (FH)
Matthias Wernicke**

studierte Industrieelektronik an der FH Ulm. Seit Anfang 2000 arbeitet er bei Vector Informatik und ist heute für das Produktmanagement der DaVinci Tool Suite verantwortlich.
matthias.wernicke@vector-informatik.de



**Dipl.-Ing. (FH)
Jochen Rein**

studierte Technische Informatik an der Fachhochschule Esslingen und ist seit 1997 bei Vector tätig. Er ist Teamleiter in der Abteilung „Software Components“ und verantwortlich für den Bereich Produktmanagement AUTOSAR BSW und CANbedded.
jochen.rein@vector-informatik.de