

AUTOSAR Migration Challenge

Zur Migration bestehender Anwendungen
Harald Nahrstedt

Die automobile Software-Entwicklungsgemeinschaft AUTOSAR hat mit der Version 2.1 einen Standard definiert, der inzwischen eine feste Größe in der Branche darstellt. Aus modellbasierter Sicht sind Architektur-Strukturen entstanden, auf deren Grundlage wieder verwendbare Steuergeräte entwickelt werden können. Ebenso sind die Vorteile einer Migration bestehender Anwendungen deutlich absehbar.

Situation

Anwendungen laufen heute oft auf mehreren Software-Plattformen und besitzen daher unterschiedliche Schnittstellen und Inhalte in ihrer Architektur. Daraus resultiert ein erheblicher Mehraufwand an Entwicklung, Tests, Wartung und Pflege. Oft liegen auch noch verschiedene Plattform-Versionen vor. Genau hier setzt AUTOSAR auf.

Derzeit werden unter anderem Basis-Plattformen entwickelt, von denen die nächsten Fahrzeuggenerationen abgeleitet werden. Dahinter steht das Ziel, ein und dasselbe Steuergerät in viele Fahrzeuge zu integrieren und damit die Kosten zu reduzieren. Gleichzeitig sollen Qualität und Stabilität der Fahrzeugelektronik erhöht werden, woraus sich ein Spagat zwischen den Unwägbarkeiten einer neu eingeführten Technik und der Stabilität des Produktes ergibt. Den häufig geäußerten Befürchtungen, durch AUTOSAR würde der Bedarf an Rechenzeit und Speicherplatz im Steuergerät wachsen und somit die Herstellungskosten steigen, stehen zahlreiche Vorteile auf der Architekturseite entgegen.

Nur durch die rechtzeitige Beschäftigung mit AUTOSAR als Architektur und Entwicklungsprozess ist es möglich, die Chancen und Risiken im jeweiligen Kontext objektiv zu bewerten, einen Migrationspfad für den Umstieg der Steuergeräteentwicklung hin zu AUTOSAR zu entwickeln und diesen dann erfolgreich umzusetzen. Dieser Übergang will gut geplant sein und sollte sinnvoll schrittweise erfolgen.

Das Ziel

Das Ziel einer hardwareunabhängigen Anwendungsentwicklung erreicht AUTOSAR zum einen durch die strikte Einführung von Layer-Strukturen (Bild 1) und zum anderen durch eine konkrete Definition der Basis-Plattform. Durch standardisierte und konfigurierbare Module soll eine Anpassung der Anwendung an unterschiedliche Software-Plattformen überflüssig werden. Layer kennt man bereits aus der Einordnung von Softwarekomponenten in größeren Projekten. AUTOSAR verwendet ein solches Layer auch als „Organisations-Manager“.

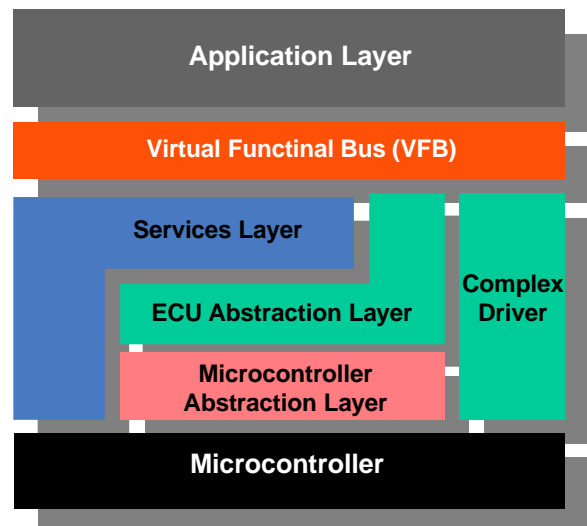


Bild 1: Das Layer-Konzept von AUTOSAR

Der VFB

Dieses Layer, von AUTOSAR als Virtual Functional Bus (VFB) bezeichnete, liegt zwischen den Anwendungen und der Hardware Abstraktion. Es „managet“ die Kommunikation (Bild 2) zwischen unterschiedlichen Anwendungen oberhalb des VFB, sowie deren Kommunikation mit der Basis-Software. Mit der Definition der Kommunikations-Ports ist eine frühzeitige Entwicklung einer Anwendung möglich, ohne die Einbindung in das Gesamtkonzept zu kennen. Die exakten Signalfade werden später per Konfiguration definiert. Nach diesem Konzept lassen sich Funktionen innerhalb der Fahrzeug-Topologie verschieben.

Die RTE

Mit der Konfiguration entsteht eine Instanzierung des VFB innerhalb einer ECU, die AUTOSAR als Runtime Environment (RTE) bezeichnet. Welche Anforderungen eine Anwendung an die RTE stellt, wird in standardisierter Form in Software Komponenten-Beschreibungen (Software Components Description) festgelegt. Diese Beschreibungen sind die Basis der Gesamtkonfiguration. Durch die RTE bleiben tiefer gelegene Software-Schichten verborgen und die Ressourcen des Betriebssystems bleiben unberücksichtigt.

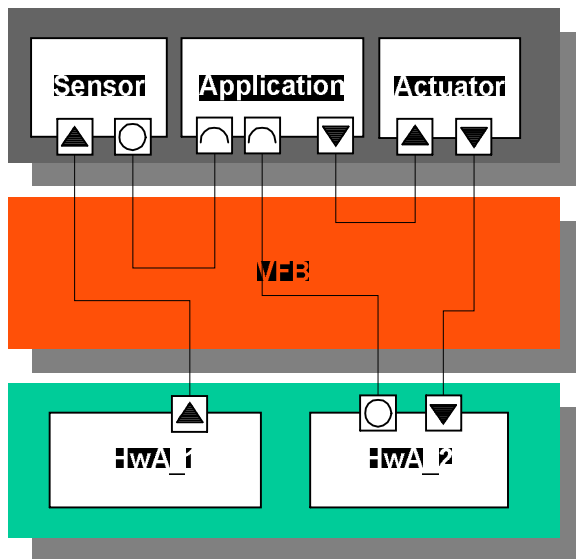


Bild 2: Die abstrakte Kommunikationsschicht

Dazu gehören zum einen die Treiberschichten, die Zugriff auf Peripherie und Kommunikation ermöglichen, aber auch die Zugriffe auf das Betriebssystem. Der Anwendungsentwickler soll nicht mehr direkt auf Ressourcen des Betriebssystems zugreifen. Außerdem wird es nicht mehr notwendig, Tasks zu definieren und deren Aktivierung und Deaktivierung zu steuern.

Runnablees

Die Laufzeitkomponenten einer Anwendung werden frühzeitig durch Runnable Entities dargestellt. Die zweite wichtige Aufgabe der RTE ist es, diese Laufzeitkomponenten zu verwalten. Mit der Konfiguration der RTE werden die Runnables auf Betriebssystem-Tasks aufgeteilt.

Die Runnable Entities bilden zusammen mit den Kommunikationsports die einzigen

Schnittstellen zwischen einer Anwendung und dem Gesamtsystem.

Generierung der Basis-Software

Aus diesem Konzept ergeben sich unter anderem die Kommunikationspfade innerhalb der ECU. Daraus können dann in einem Integrationsschritt die Zugriffe auf die erforderlichen standardisierten Basis-Software-Module konfiguriert und generiert werden (Bild 3). Dieser Schritt ist für den Einsatz der Basis-Software entscheidend. Einige Parameter sind architekturenspezifisch und können von einem „Integrator“ erst in dieser Phase bestimmt werden. Dabei können sich wiederum neue Erkenntnisse ergeben, die dann unter Umständen eine erneute Konfiguration auf Systemebene erfordern.

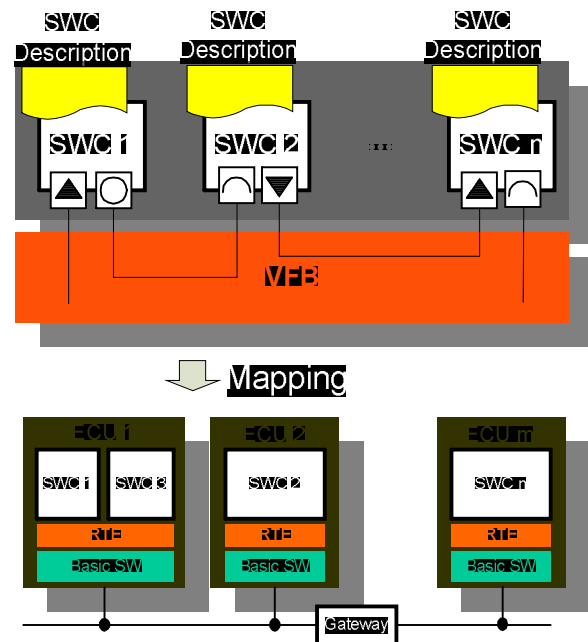


Bild 3: Das Konfigurationskonzept

Beispielsweise kann die „Verfeinerung“ des Scheduling ergeben, dass die Randbedingungen einer Runnable Entity nicht erfüllt werden können und somit diese Software-Komponente auf dieser ECU nicht lauffähig ist.

Zum Einsatz der Basissoftware in aktuellen Steuergeräten lassen sich zwei gegensätzliche Ansätze feststellen. Zum einen geben OEMs Softwarekomponenten vor oder zumindest deren Spezifikation, zum anderen sind die Steuergerätehersteller bestrebt, hausintern immer die gleiche Architektur für eine Plattform zu verwenden.

Migration bestehender Anwendungen

Der Einsatz von AUTOSAR in der Steuergeräteentwicklung hat erhebliche Auswirkungen auf Hardware, Software und den Entwicklungsprozess. Es sollte frühzeitig mit der Evaluierung von AUTOSAR begonnen werden. Der Umgang mit der standardisierten Basissoftware, der „Mechanismus“ RTE als Kommunikations- und Organisations-Struktur und die damit zusammenhängenden Entwicklungs- und Generierungsprozesse können nicht früh genug untersucht werden.

Der Weg von der bisherigen steuergeräteorientierten hin zu einer funktionsorientierten Entwicklung mit AUTOSAR erfolgt typischerweise über ein Pilotprojekt. Auf der Grundlage eines vorhandenen Steuergeräts wird die bestehende Funktion auf die AUTOSAR-Architektur umgestellt und somit untersucht, welche Auswirkungen AUTOSAR auf die Leistungsfähigkeit des Systems im Speziellen und auf den Hard- und Softwareentwicklungsprozess im Allgemeinen hat. Dazu ist der Einsatz AUTOSAR kompatibler Basissoftware unerlässlich. Die besten Erfahrungen liefert im „ersten Go“ eine „einfache vertikale“ Entwicklung (Bild 4). Sie kann dann mit den gewonnenen Erfahrungen „waagerecht“ ausgeweitet werden.

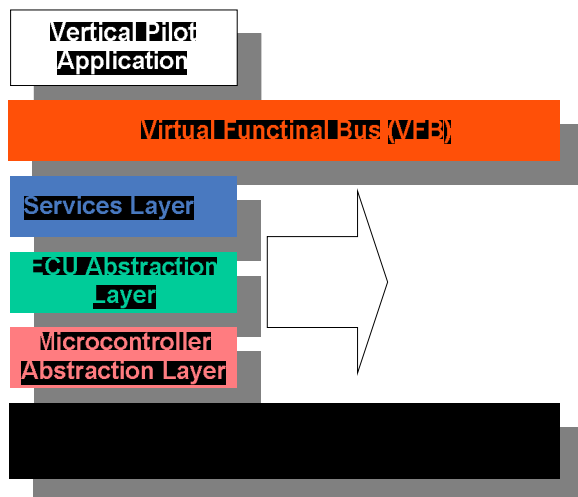


Bild 4: Vertikaler Prototyp

Mit der Umgestaltung vorhandener Anwendungen können Integrationsentwickler erste Erfahrungen mit der AUTOSAR Basissoftware und deren Konfiguration sammeln und dabei die grundlegenden Methoden von AUTOSAR erlernen. Wenn die Methode erst einmal vollständig angewendet wird, ist dies nicht mehr möglich, da diese Verfahren durch die

Auftrennung der Rollen und deren Generierung dann zum Teil nicht mehr sichtbar sind.

Für die Migration bestehender Anwendungen ist von großem Vorteil, dass die verwendeten Funktionen bekannt sind und als Ausgangsbasis dienen können. Eine Anforderungsanalyse kann in der Regel entfallen. Es lassen sich sogar Teile der Software wieder verwenden. Außerdem bietet sich vor allem die Chance, im Zuge eines ReDesigns vorhandener Steuergeräte, klarere Strukturen in die Software einzuziehen. Erweiterungsfähigkeit und Wartbarkeit werden erhöht.

Für die Restrukturierung in Richtung AUTOSAR gilt es vor allem, den bestehenden Code in Applikationscode und Basissoftware zu unterteilen. Anwendungen, die heute schon eine klare Schnittstelle zu den Peripherie- und Kommunikationsmodulen haben, sowie eine unkomplizierte, möglichst unverzahnte Verbindung zum Betriebssystem verwenden, lassen sich leicht in AUTOSAR Application Software Components weiterentwickeln. Möglicherweise können Teile der Anwendung wieder verwendet werden bzw. Änderungen auf die Schnittstellen beschränkt bleiben. Zur vollständigen Verwendung der Application Software Component in AUTOSAR-Systemen wird dann noch die Beschreibung der Komponente in einem AUTOSAR-Format auf der Basis von XLS benötigt.

Zur Umstellung der eigenen Anwendungssoftware ist es nicht unbedingt erforderlich, schon auf vollständigen AUTOSAR-Basis-Software-Modulen aufzusetzen. Wichtiger ist es, innerhalb der eigenen Anwendungssoftware die Schnittstellen zur AUTOSAR RTE zu verwenden und die damit verbundenen grundsätzlichen Designregeln zu beachten. Bei der Verwendung der RTE API (Run-Time Environment Application Program Interface) ist die Semantik der Schnittstelle entscheidender als eine perfekte Syntax - jedenfalls im ersten Ansatz.

Zu den Design-Entscheidungen gehört zum Beispiel auch die Frage, ob die Runnable Entities einer Software-Komponente zyklisch oder per „Data Received Event“ eines zugehörigen Empfangsports getriggert werden können. Durch die existierende Kommunikationsmatrix sind Zykluszeiten bereits teilweise vorgegeben und Event-Signale, die

eine Software-Komponente senden muss, kommen auch möglicherweise aus anderen Steuergeräten.

Ein Zwischenschritt kann auch die Schaffung einer weiteren Abstraktionsschicht zwischen RTE und bestehender Basis-Software sein, um so die vorhandene Basissoftware auf der bestehenden Hardware weiter verwenden zu können.

Eine weitere Erfahrung bei der Umgestaltung ist die Generierung der Implementierung aus modellbasierten Entwicklungswerkzeugen. Hierbei wird dem Tool-Lieferanten die korrekte Umsetzung der Verwendung der RTE-Schnittstellen überlassen. Auch hier lassen sich noch Erfahrungen sammeln. Als Entwicklungswerkzeug sollte für alle Module ein einheitlicher „Konfigurator“ gefunden werden, so dass auch die Konsistenz der Konfiguration gewährleistet ist.

Optimierung des Quellcodes

Nach der Generierung des Quellcodes stellt sich ein weiteres Problem, die Optimierung hinsichtlich Speicherbedarf und Laufzeit. Dass AUTOSAR mehr Ressourcen benötigt liegt an der Tatsache, dass zum einen die Funktionalität durch weitere Module erhöht wurde und zum anderen Funktionen aus den Anwendungen sich nunmehr in der Plattform wieder finden. Zur Optimierung betrachtet man zunächst die einzelnen Module und erst danach Modulgruppen bezüglich Datenkonsistenz und Laufzeitverhalten von Signalen. So lassen sich signifikante Optimierungen erzielen.

Standardisierungsgrad

Hinzu kommt, dass der Standardisierungsgrad der Software nicht so umfassend ist, wie ihn AUTOSAR beschreibt. Hier gilt es, einen Standard der Software zukünftig nur in den notwendigen Grenzen einzuführen, um Raum für neue Innovationen zu lassen.

Zukunft

Durch die Komplexität von AUTOSAR erscheint der neue Standard auf den ersten Blick schwierig umsetzbar und erlernbar. Wenn AUTOSAR jedoch zum Standard für die Automotive-Software-Entwicklung wird, dann eröffnet dies eine völlig neue Dimension.

AUTOSAR wird entscheidend zur Qualität und Innovation zukünftiger Systeme beitragen. Eine ganzheitliche AUTOSAR-Standard-Core-Lösung mit einem durchgängigen Konfigurationskonzept erleichtert den Weg in diese neue Architektur der Automotive-Elektronik.

Der Anwendungsentwickler muss dabei in kommunizierenden Software-Komponenten denken statt in Tasks und Zeitscheiben. Das Betriebssystem bleibt für ihn unsichtbar und die komplette Anwendung muss über die RTE mit der Basis-Software verbunden werden. Gleiches gilt für alle Treiber und die anderen Software-Schichten unterhalb der RTE. Diese Veränderungen werden zunächst gewöhnungsbedürftig sein. Auf lange Sicht wird sich diese Methode aber später bewähren, da sie nahezu perfekt zum Konzept der objektorientierten Software-Entwicklung passt.

Tools

Für zukünftige Tools gilt, dass sie die AUTOSAR-Schnittstellen bedienen können und dabei trotzdem für die Integration von Drittkomponenten offen bleiben. Vor allem Konfigurationswerkzeuge sollen diese Herausforderung meistern und den Entwickler auch bei der Validierung seiner Strukturen unterstützen.

Die Tools rund um AUTOSAR lassen sich in die drei Kategorien Design, Konfiguration und Test/Simulation aufteilen. Insbesondere geeignete Testinstrumente sind wichtiger Bestandteil einer erfolgreichen Entwicklung. Ein Steuergerät arbeitet als Teil eines Ganzen. Um die Konsistenz im Gesamtsystem zu überprüfen und sicherzustellen, ist ein leistungsfähiges Simulationswerkzeug erforderlich.

#hn