



Adaptive AUTOSAR – Eine neue Art des Arbeitens

Im vorliegenden Beitrag werden einige Themen vorgestellt, die beim Umgang mit der neuen serviceorientierten Architektur (SOA) zu beachten sind. Eine auf Adaptive AUTOSAR basierende SOA ersetzt die klassische AUTOSAR-Plattform nicht 1:1. Es soll hier lediglich gezeigt werden, dass sowohl im technischen Bereich als auch hinsichtlich der Arbeitsprozesse neue Herausforderungen entstehen.

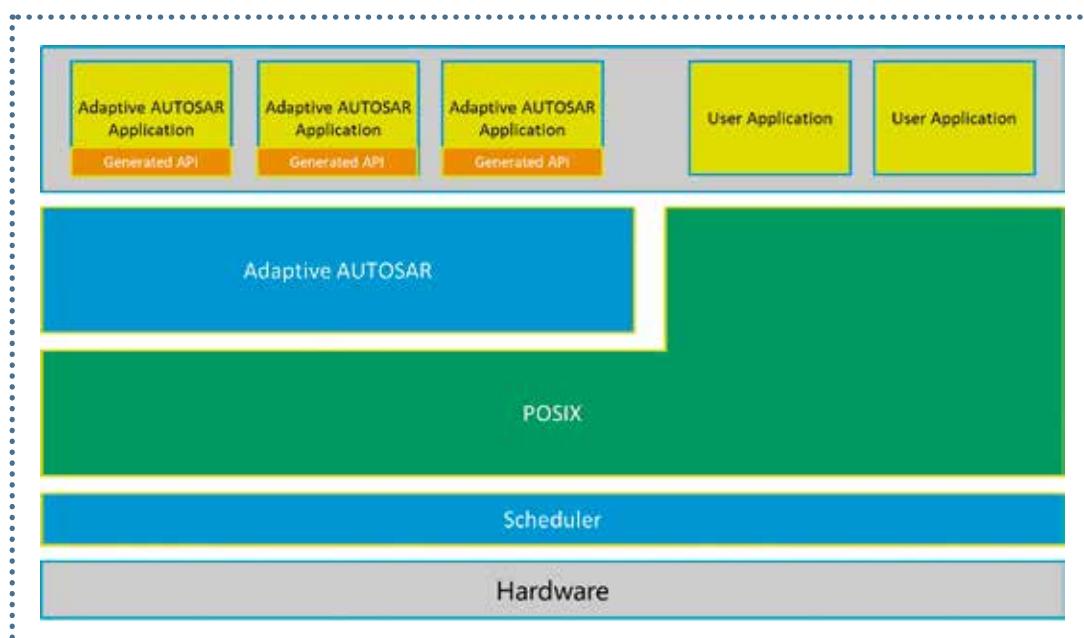


Bild 1: SOA mit Adaptive AUTOSAR. (© KPIT)

Die Softwareplattformen, die heute in der Automobilindustrie vorwiegend genutzt werden, bauen auf Firmware auf. Die Firmware ist ein binär laufender Build für Bare-Metal in einem bestimmten Mikrocontroller. Es wird ein Anforderungskatalog erstellt und als Firmware-Implementierung oder -Konfiguration realisiert. Sie wird bei der Produktion, einer regelmäßigen Wartung in einer Werkstatt oder im ungünstigsten Fall bei einem Rückruf aufgrund von enthaltenen Fehlern heruntergeladen. Das ursprüngliche Ziel besteht jedoch darin, dass der Hersteller nach dem Download niemals einen Software-Update während der Lebensdauer des Fahrzeugs vornehmen muss.

Diese Vorgabe passt nicht sehr gut zu Software. Nicht weil die Automobilindustrie schlechte Softwareingenieure hätte. Aber die Funktionen und Aufgaben, die elektronische Steuergeräte heute übernehmen müssen, sind so komplex, dass sich kontinuierliche Funktionsupdates sowie auch Softwarefehler nicht vermeiden lassen. Je komplexer ein Projekt

ist, desto stärker steigt die Zahl der Fehler pro Codezeile an [Code2], d.h. die Anzahl der Fehler ist nicht nur eine lineare Funktion der Codezeilen. Es können natürlich größere Investitionen in Überprüfung und Validierung vorgenommen werden, aber dem Funktionswachstum wird oft Vorrang vor der Modul- und Integrationserprobung gegeben. Zusätzlich verlangt eine typische Überprüfung das gleiche Expertenniveau wie bei der Implementierung. Der Systemtest ist teuer und schwierig zu koordinieren. Außerdem sind die Anforderungen nicht in allen Fällen von Beginn an klar, was die Validierung des Systems erschwert.

Die neue Generation der Fahrzeugeigentümer erwartet neueste und moderne Merkmale im Auto. Die meisten dieser Funktionen stehen in Verbindung mit der Software. Smartphones sind heute weit effizienter in der Anpassung an die Benutzeranforderungen. Sie sind permanent mit dem Internet verbunden und lösen viele Aufgaben, die als normaler Bestandteil der heutigen Lebensweise betrachtet werden.

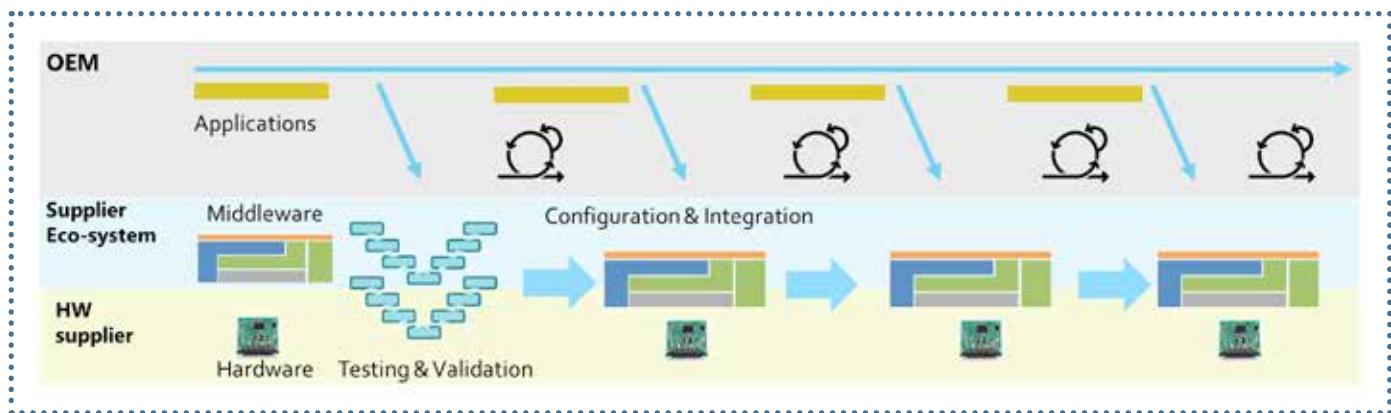


Bild 2: Continuous Integration. (© KPIT)

Bei Infotainment- und Telematikprojekten wurden schon früh komplexe Funktionsimplementierung und ein agiler Arbeitsprozess eingeführt. Üblicherweise handelt es sich dabei um aktualisierbare Systeme mit einem Dateisysteme unterstützenden komplexen Betriebssystem wie Linux. Leider stellen diese Projekte in vielen Fällen eine Integration zahlreicher Module dar, wodurch eine Softwareplattformintegration entsteht, die sich nur wenig oder gar nicht zur Wiederverwendung eignet. Jede Generation eines Projekts ist im Grunde ein Neubeginn. Darüber hinaus sind Linux-basierte Plattformen nicht sehr gut darin, Sicherheits- und anspruchsvolle Echtzeitanforderungen zu erfüllen.

Adaptive AUTOSAR

Vor diesem Hintergrund ist Adaptive AUTOSAR ein neuer Standard, der die oben erwähnten Probleme löst und eine teilweise service-orientierte Architektur (SOA) schafft.

Classic AUTOSAR ist eine eigenständige Firmware-Plattform, während Adaptive AUTOSAR nicht versucht, in diesem Sinne vollständig zu sein, um eine komplette SOA zu bilden. Eine SOA ist eine Softwareplattform, bei der Dienstleistungen im Mittelpunkt stehen. Eine Dienstleistung wird von einem Dienstleister erbracht und von einem Dienstleistungsnutzer in Anspruch genommen. Die Verbindung zwischen einem Dienstleister und einem Dienstleistungsnutzer wird von einem Service-Broker dynamisch zur Laufzeit hergestellt. Der Dienstleistungsnutzer muss nicht unbedingt wissen, wo sich der Dienstleister befindet. Daher ist die Softwareplattform dynamisch und kann teilweise ohne Auswirkungen auf das System aktualisiert werden.

Wie oben erwähnt, beansprucht Adaptive AUTOSAR nicht, eine komplette SOA zu schaffen. Das Betriebssystem ist nur verknüpft und nicht spezifiziert [AUTOSAR OS]. Hier verbindet sich eine Adaptive AUTOSAR-Anwendung mit einem Betriebssystem, das die API-Untergruppe PSE51 des POSIX-Standards bereitstellt [POSIX]. Dies bedeutet nicht, dass das Betriebssystem nur PSE51 sein kann, sondern dass die Adaptive AUTOSAR-Anwendung nur diese API verwenden kann. Das Betriebssystem kann immer noch ein vollständiges PSE54-System sein (z.B. das PSE54-basierte Linux). Dieses Konzept hat den Vorteil, dass alle POSIX-konformen Anwendungen und Bibliotheken modelliert werden können.

Zudem können sie auf einer beliebigen AUTOSAR-Softwareplattform wiederverwendet und eingesetzt werden. In anderen Worten, AUTOSAR versucht nicht, POSIX neu zu definieren. Nur zur Klärung sei daran erinnert, dass POSIX eine Standardschnittstelle für eine Programmoberfläche ist. Es ist weder eine implementierte Anwendung noch ein Betriebssystem (Bild 1).

Es gibt eine Nachfrage nach alter zuverlässiger Software, die schon lange angeboten wird, da befürchtet wird, dass eine neue Software immer Fehler enthält. Warum nicht also einfach den alten Code verwenden und einige kleinere Erweiterungen vornehmen? Die heute in der Software verlangten Funktionen entwickeln sich sehr stark weiter. Es ist sehr schwer, eine gute Architektur zu schaffen, die es ermöglicht, alle Arten von künftigen Merkmalen hinzuzufügen, die nicht von Anfang an bekannt sind. Ein Refactoring der Software ist von Zeit zu Zeit erforderlich, um die Software effizient pflegen und überprüfen zu können. In vielen Fällen wird so in Bezug auf das klassische AUTOSAR argumentiert. Der Vorzug von Adaptive AUTOSAR liegt darin, dass POSIX-konforme Software wiederverwendet werden kann, nach der Philosophie „Einmal schreiben, überall anpassen“. Es besteht kein Zwang, etwas umzuschreiben und können von bestehender offener Software profitieren, oder ein Software-Unternehmen kann Anwendungen für unterschiedliche Kunden wiederverwenden.

Die Entscheidung für eine service-orientierte Architektur (unter Verwendung von Adaptive AUTOSAR) beseitigt nicht alle Hindernisse auf dem Weg. Der Umstieg von einem firmwarebasierten Ansatz (d. h. Classic AUTOSAR) auf SOA bedeutet, dass man vor neuen Herausforderungen steht, die bisher nicht bekannt waren oder in erster Linie manuell gelöst wurden. Einige davon werden hier behandelt.

Eine aktualisierbare SOA impliziert, dass sich der Arbeitsprozess der Projekte verändern muss. Das klassische Wasserfallmodell, bei dem ein Lieferant eine Anforderungsliste erhält und diese umsetzt und testet, funktioniert dann nicht mehr wirklich. Stattdessen ist eine agile Arbeitsweise auf der Grundlage eines Continuous Integration (CI)-Konzepts erforderlich (Bild 2), nämlich eine CI, die kontinuierlich neue Software überprüft und freigibt. Zusätzlich verbinden sich der OEM und Lieferanten mit der CI, und die Lieferanten haben eine eigene CI für ihre Anwendungen oder Softwareplattfor- »



men. Um einem solchen CI-Ökosystem gerecht zu werden, müssen die Cls von OEM und Lieferanten automatisch interagieren. Cls, die automatisch Versionen für die CI anderer Unternehmen (OEMs) erstellen, sind für IT- und Rechtsabteilungen eine schwer zu knackende Nuss. Man muss die Eigentumsstruktur verstehen und wissen, wie sich solche Interaktionen sicher durchführen lassen.

Wiederverwendbarkeit

Ein weiteres wichtiges Thema ist es, eine Softwareplattform für verschiedene Projekte wiederverwendbar zu machen, ohne jedes Mal von Null zu beginnen. Hier erweist sich der Adaptive AUTOSAR-Standard als sehr hilfreich. Einerseits gibt es genau umrissene AUTOSAR-Basisdienste wie Diagnostic Manager für spezifische im Fahrzeug erforderliche Funktionen und andererseits eine klare Methodik und ein Austauschformat zur Verwendung durch OEMs und Lieferanten im Arbeitsprozess. Die Existenz eines solchen Standards ermöglicht die Entwicklung und Wiederverwendung von Anwendungen für unterschiedliche Projekte und Kunden.

Der Vorteil eines Firmware-Konzepts besteht darin, dass Updates einmalige Vorgänge sind. Bei einem aktualisierbaren System werden hingegen räumlich und zeitlich unterschiedliche Teil-Updates vorgenommen. Das Basis-Image wird implementiert, und verschiedene Organisationen (und Unternehmen) führen zu unterschiedlichen Zeiten Updates am System durch. Dies stellt hohe Anforderungen an die Bereitstellung von Offline-Tools. Das Update muss vor der Implementierung auf Kompatibilität überprüft werden. Das Update ist so zu verifizieren, dass es zu keinerlei Ausfällen im Fahrzeug führen wird. Auch müssen Updates orchestriert werden, wenn z.B. zwei elektronische Steuereinheiten aktualisiert werden müssen und eine ausfällt, muss die Software in beiden ECUs auf die Vorgängerversion zurückgesetzt werden.

Service-orientierte Architektur

Der POSIX-Standard definiert verschiedene Merkmale, wobei die Feature-Ebene von den PES51-54-Profilen gesteuert wird. In jedem Profil sind die meisten Merkmale optional [POSIX]. Im klassischen AUTOSAR ist der Funktionsinhalt klar umrissen und wird meist von den Skalierbarkeitsklassen gesteuert. Zwei verschiedene POSIX-Betriebssysteme können sich in ihrem Funktionsinhalt sehr stark unterscheiden. Es ist wichtig, die benötigten Merkmale eines POSIX-Betriebssystems in einer frühen Projektphase festzulegen und zusätzlich sicherzustellen, dass das gewählte POSIX-Betriebssystem mit weiteren Merkmalen erweiterbar ist, wenn das Projektwachstum dies erfordert.

SOA wird oft mit dynamischem Verhalten in Verbindung gebracht. Aber um die Software der nächsten Generation in Fahrzeugen steuern und verwalten zu können, muss der Arbeitsprozess stringent und automatisiert sein. Außerdem muss die Software im Fahrzeug gesteuert werden, um Verifizierung und Sicherheit zu erreichen. Lieferanten und OEMs für Fahrzeuge werden enger zusammenarbeiten und dabei

automatisierte kontinuierliche Integrationssysteme nutzen. Die statische Prüfung ist durch eine dynamische Prüfung im Fahrzeug zu erweitern (z.B. durch Hinzufügen von Metadaten). Unterschiedliche Lieferanten im Ökosystem der Softwareplattform müssen effizient zusammenarbeiten. Dazu ist ein vertragsbasiertes Entwicklungskonzept vonnöten.

Dies sind einige Themen, die beim Umgang mit dem neuen SOA-Modell zu beachten sind. Eine auf Adaptive AUTOSAR basierende SOA ersetzt die klassische AUTOSAR-Plattform nicht 1:1. Es sollte hier nur gezeigt werden, dass es sowohl im technischen Bereich als auch hinsichtlich der Arbeitsprozesse neue Herausforderungen zu bestehen gibt. Der Vorteil des Konzepts von Adaptive AUTOSAR ist die Angleichung an einen bestehenden Standard wie POSIX. Daher ist es möglich, Software und Erfahrungen aus verschiedenen Projekten (auch aus anderen Bereichen als der Automobiltechnik) wiederzuverwenden. ■ (oe)

» www.kpit.de

Referenzen

- [Code2] Code Complete, 2. Ausgabe, Steve McConnell
- [AUTOSA ROS] Spezifikation der Betriebssystemschnittstelle
- [POSIX] <http://pubs.opengroup.org> und IEEE 1003

» www.hanser-automotive.de/6437064

Hier finden Sie die Download-Version des Beitrags.



Anders Kallerdahl ist Subject Matter Expert Automotive Software, bei KPIT Technologies