



AUTOSAR-Komponenten modellieren

Durchgängig modellbasierte Entwicklung auf allen Ebenen



Mathias Langer, Arne Noyer, Günter Kircher, Gert Bikker, Elke Pulvermüller
Institut für Verteilte Systeme, Ostfalia Hochschule Wolfenbüttel

Durch die zunehmende Komplexität von eingebetteten Systemen – insbesondere im Kraftfahrzeugbereich – steigt das Interesse an Modellierungswerkzeugen im Entwicklungsprozess. Mit AUTOSAR hat sich ein Standard etabliert, der ebenfalls eine Notation zur modellbasierten Entwicklung beinhaltet. Dieser beinhaltet darüber hinaus noch weiteres, wie z.B. eine Runtime Environment. Da es für diese Aspekte unterschiedliche Werkzeuge gibt, müssen zwischen diesen stets Daten ausgetauscht werden. Dafür gibt AUTOSAR das ArXML-Format vor. Der Beitrag behandelt, wie Komponenten im AUTOSAR-Umfeld modelliert werden. Anschließend werden aus dieser Beschreibung ArXML-Dateien erzeugt, die Informationen über die Komponenten beinhalten. Die Komponenten-Beschreibung enthält jedoch noch nicht das Verhalten einer zu entwickelnden Software. Daher wird diskutiert, wie die Komponenten in UML überführt werden können, damit anschließend das Softwareverhalten mit UML ebenfalls modellbasiert definiert werden kann.

Einleitung

Heutzutage sind elektronische Steuergeräte, Sensoren, Aktoren und Bussysteme ausschlaggebende Komplexitätstreiber und Kostenfaktoren bei der Entwicklung von Fahrzeugen aber auch bei anderen Systemen. Es gibt eine ganze Reihe an Faktoren die zur Softwarekomplexität im Fahrzeug beitragen, wie dwaregetriebene Softwareentwicklung, Softwareabhängigkeiten, verteiltes Entwickeln der Komponenten oder der Lebenszyklus von Hard- und Softwarekomponenten [1]. So führt die hohe und weiterhin zunehmende Anzahl an Funktionen zum Beispiel zu einem stark anwachsenden Kommunikationsaufwand der Komponenten eines Fahrzeuges. Die Software hat somit eine Komplexität erreicht, welche nur sehr schwer beherrschbar ist.

Um diese Systemkomplexität und die steigende Anzahl von Abhängigkeiten zu bewältigen und dabei gleichzeitig die Kosteneffizienz zu verbessern, mussten die Schnittstellen zwischen Kommunikationsbussystemen, Anwendungen und der Basis-Software standardisiert werden. Hierfür wurde von namhaften Automobilherstellern, Zulieferern und Toolherstellern die AUTOSAR- Initiative gegründet [3], durch welche ein gemeinsamer Standard für die Spezifikation und Implementierung von Software in der Automobil-industrie etabliert wurde.

Da der AUTOSAR-Standard sehr ngreich und komplex ist, ist das Ausbilden von Spezialisten, welche mit diesem Standard umgehen können, sehr kompliziert, t- und kostenintensiv. Das ist sehr wichtiger Grund Wissen in diesem Bereich bereits in Rahmen des Studiums zu vermitteln. Dabei ist es sehr hilfreich AUTOSAR nicht nur in der Theorie sondern auch mit der Hilfe eines kleinen praktischen Beispiels kennenzulernen.

Stand der Technik



AUTOSAR (AUTomotive Open System ARchitecture) ist ein internationaler Standard der Automobilindustrie. Dieser Standard wurde entwickelt, um eine gemeinsame Referenzarchitektur, an Stelle bisher üblichen Standardarchitekture eines Herstellers, zur Softwareentwicklung in einem Automobil zu definieren. AUTOSAR soll gemeinsame Rahmenbedingungen für die Entwicklung und den Austausch sowohl von Software- als auch Hardware-Komponenten derartiger Systeme schaffen. Der AUTOSAR Standard definiert nicht nur eine umfassende technische Infrastruktur für diese Systeme, sondern auch eine darauf aufbauende Methodik und Beschreibungsformate für die Entwicklung AUTOSAR konformer Systemspezifikationen.



Hinter dem Akronym AUTOSAR verbirgt sich eine umfassende Reihe von Spezifikationen, die folgende Aspekte enthalten [6]:



- Architektur – Software-Architektur einschließlich AUTOSAR Basis Software(BSW) als Integrationsplattform für Hardwareunabhängige Software-Anwendungen.
- Methodik – Austauschformate oder Beschreibungsvorlagen ermöglichen eine nahtlose Konfiguration der BSW und die Integration von Anwendungssoftware in Steuergeräten. Das beinhaltet Methodik, wie dieses Gerüst verwendet wird.
- Applikation Interface – Spezifikation der Schnittstellen der typischen Automotive-Anwendungen aus allen Bereichen im Hinblick auf die Syntax und Semantik sollten als Standard für die Anwendungssoftware dienen.




Die Ziele von AUTOSAR sind dabei die Wiederverwendung von Komponente






- Verwaltung von zunehmenden E/E(Elektronik / Elektrik)-Komplexität - Im Zusammenhang mit dem Wachstum im Funktionsumfang.
- Verbesserte Flexibilität – mehr Raum für Updates, Upgrades und Modifikationen
- Verbesserte Skalierbarkeit – das System kann in einer einfachen Weise vergrößert werden.
- Verbesserte Qualität und Zuverlässigkeit – die Wiederverwendung von bewährten Software-Anwendungen und Erkennung von Fehlern in frühen Entwicklungsphasen.

Der entwickelte AUTOSAR-Standard befindet sich zurzeit bereits in einer weit vorgeschrittenen Phase. Die Beschreibung besteht schon aus mehr als 9000 Seiten Dokumentation und die ersten Serien-Steuergeräte wurden nach AUTOSAR-Architektur implementiert.

 AUTOSAR-Methodik (AUTOSAR-Methodology) ist eine standardisierte Beschreibung des Entwicklungsprozesses von AUTOSAR-konformen Projekten. Dieser Prozess ist in mehrere Phasen unterteilt. Für die einzelnen Phasen werden die Werkzeuge von unterschiedlichen Herstellern eingesetzt. Der Austausch von Zwischenprodukten nach jeder Phase ist durch den Einsatz von speziellen standardisierten Dokumenten im XML-Format (ArXML) möglich. Zu diesen Dokumenten gehören unter anderem:

- Software Component Description
- ECU Resource Description
- System Constraint Description

Die Beschreibungen  Zeitschiene, Iterationsschritten oder beteiligten Rollen sind kein Bestandteil der AUTOSAR-Methodik.  müssen für jedes Unternehmen oder Projekt spezifisch definiert werden. Grundsätzlich kann man die AUTOSAR-Methodik in drei Entwicklungsphase  Systemkonfiguration, ECU-Konfiguration und Komponentenentwicklung unterteilen. Diese können wegen der standardisierten Austauschformate auch parallel bearbeitet werden.

Lange oder kurze Fassung wählen

Die AUTOSAR-Methodik für die Entwicklung von AUTOSAR konformen Projekten ist in die drei großen Entwicklungsbereichen, Systemkonfiguration, ECU-Konfiguration und Komponentenentwicklung unterteilt, welche mit speziellen Werkzeugen abgedeckt werden müssen. Die Beschreibungen zu Zeitschiene, Iterationsschritten oder beteiligten Rollen sind kein Bestandteil der AUTOSAR-Methodik. Dies müssen für jedes Unternehmen oder Projekt spezifisch definiert werden.

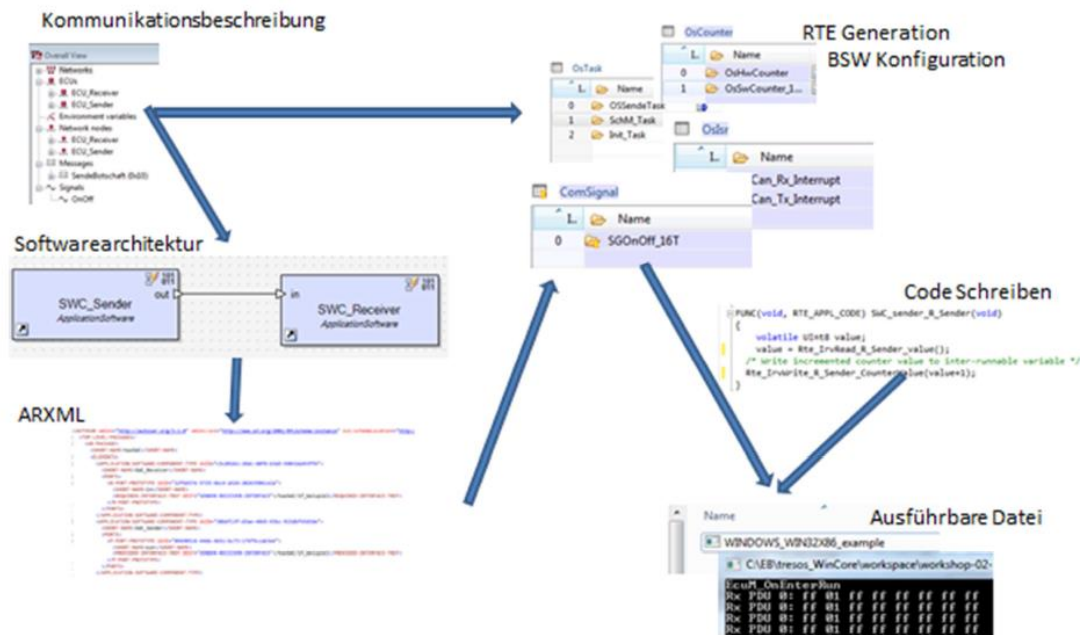


Abbildung 1: Workflow für die Entwicklung von AUTOSAR-konformen Software

Ein möglicher Workflow für die Entwicklung AUTOSAR-konformer Applikationen besitzt die folgenden Entwicklungs- und Konfigurations-Schritte:

- Erstellen oder Modifizieren der Kommunikationsbeschreibung z.B. mit Vector CANdb++ Editor und Speicherung in DBC, FIBEX oder LDF -Dateien
 - Import der Kommunikationsbeschreibung
 - Definition der Softwarearchitektur inklusive Mapping der SWC auf ECU und der Signale auf die Ports von SWC z.B. mit dSPACE SystemDesk
 - System-Description Datei in ARXML-Format erzeugen
 - Import von System-Description Datei und der Kommunikationsbeschreibung z.B. mit EB tresos Studio
 - Konfiguration der Basis-Software inklusive RTE
 - Implementierung der Runnables
 - Generieren und Kompilieren der Applikation
- Einzelne Atomare SWCs mit ECUs verbinden
 - Bus-Mapping für die Verbindung der Hardware-Topologie mit einem Netzwerk
 - Signal-Mapping, wodurch die Datenelemente mit den Netzwerksignalen verbunden werden und womit die Kommunikation zwischen SWCs geleistet wird. Werkzeuge

Für die einzelnen AUTOSAR-Methodik Phasen gibt es eine Vielzahl von verschiedenen Werkzeugen unterschiedlicher Hersteller.

Modellierung von AUTOSAR-Komponenten



- Entwurf von SWCs: Beschreibung von Ports, Runnables und Software-Komponenten Kommunikation
- System-Architektur Entwurf: Modellierung des gesamten Systems aus Softwarekomponenten(SWC), Steuergeräten(ECUs) und Kommunikationsnetz
- Konfiguration von Basis-Software(SW) und Runtime Enviroment (RTE)
- Die Implementierung von SWCs

Systemkonfiguration

Für die Beschreibung der elektrischen Hardware sieht der AUTOSAR Entwicklungsprozess eine Kombination aus zwei Arten von Diagrammen vor:

- ECU-Diagramm definiert die Typen und Ausstattung von Steuergeräten und ihre Schnittstellen zu Bussystemen.



- Topologie-Diagramm legt die physikalische Verteilung der Hardware im System einschließlich ihrer Verschaltung fest.

SystemDesk ermöglicht die graphische Modellierung von Softwarearchitekturen mit Softwarekomponenten, welche über die Ports miteinander verbundenen sind. Prinzipiell unterstützt SystemDesk den Modellierungsprozess mit zwei Diagrammtypen (Abbildung 3-5):

- Komposition-Diagramm. Auf diesem Diagramm wird ein Überblick des gesamten Systems mit SWCs, Ports und Verbindungen zwischen SWCs dargestellt.
- Komponente-Diagramm. Bietet mehr detaillierten Einblick auf das System, man die einzelnen SWC's und Interfaces modelliert.

In dieser Phase wird die System Description erzeugt. Hierbei werden die Softwarekomponenten mit internen Verhalten erzeugt und auf ECU verteilt. Diese Konfiguration ist Voraussetzung für die ECU-Konfiguration.

Software -, Hardware-Komponenten (Mapping) Aufteilung auf die Steuergeräte

- Softwarekomponentenbeschreibung mit allen Eigenschaften, wie z.B. Konnektoren, Interfaces, Ports
- Festlegung von Kommunikationsmechanismen und Medien über K-Matrix
- Zuordnung jeder Softwarekomponente zu einer ECU
- Systemkonfigurationsbeschreibung mit Mapping und Hardware-Topology

Ergebnis ist System Configuration Description

Steuergerätkonfiguration

Zu dieser Konfiguration werden zuerst alle notwendigen

Pro ECU oder ein Gerät

Bei der ECU-Konfiguration wird zuerst aus System Description die ECU Extrakt für genau eine ECU gewonnen. Durch diese Systembeschreibung werden dann die Teile der BSW, OS und RTE konfiguriert.

Informationen, wie z.B. Task-Scheduling, benötigte BSW-Module, hinzugefügt.

Ergebnis ECU Beschreibungsdatei mit Hardware-, BSW- und RTE-Konfiguration, die wiederum für das Generieren von Steuergerätsoftware verwendet wird.

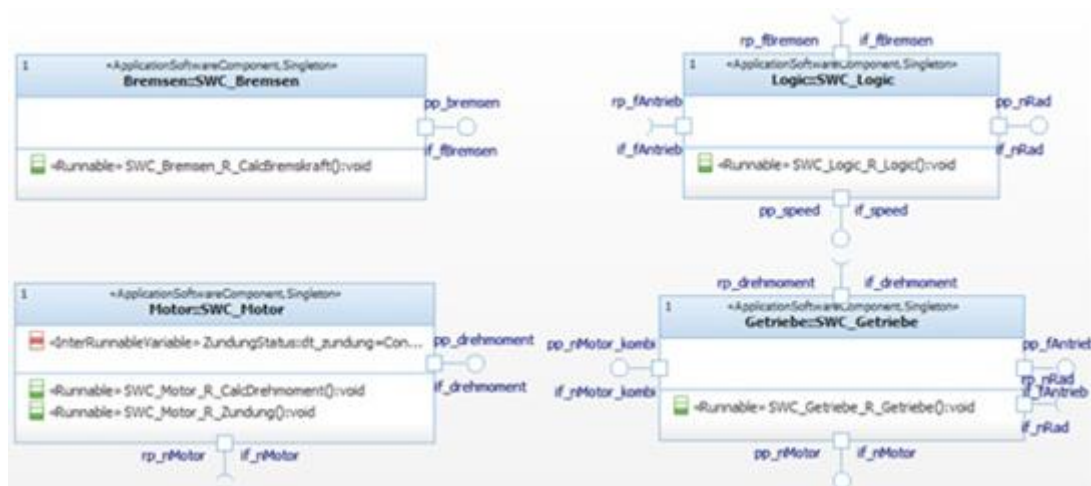
Komponentenentwicklung

Die Komponenten werden durch den Code ihrer Runnables repräsentiert und existieren nur auf der Modellierungsebene. Die Code-Entwicklung der Komponenten basiert normalerweise auf der Programmiersprache C. Die Komponenten werden in der Methodik nur in dem letzten Schritt benötigt, der das Erzeugen einer Executable-Datei für die ECU abbildet. Diese Entwicklungsphase ist aber nicht AUTOSAR-standardisiert und kann daher entwicklerabhängig realisiert werden.

Mapping zwischen Hardware-unabhängigen Softwarekomponenten und realen Steuergeräte-Hardware

Austausch zwischen Werkzeugen über ArXML-Dateien

Überführung von AUTOSAR-Komponenten in die UML und Verhaltensmodellierung von AUTOSAR-Komponenten mit der UML



Datentypen

Interfaces

Klassen

Attribute (Interrunnable Variablen)

Methoden (Runnablees)

Events

Ports

→ Verhaltensmodellierung mit Aktivitätsdiagramm

Komponenten

Zusammenfassung und zukünftige Arbeiten

AUTOSAR Standard mit Methodik,

Überblick modellbasierte Entwicklung mit dem AUTOSAR-Standard

AUTOSAR-konforme Werkzeuge

Danksagungen

Abkürzungsverzeichnis

Abbildungsverzeichnis

AUTOSAR (AUTomotive Open System ARchitecture)

Literatur- und Quellenverzeichnis

Autor

Mathias Langer arbeitet an einem Forschungsprojekt in der Ostfalia Hochschule in der Arbeitsgruppe von Prof. Dr.-Ing. Gert Bikker. Das Forschungsprojekt wird in Zusammenarbeit mit Partnern aus der Industrie für die AUTOSAR-Entwicklung durchgeführt. Weiterhin ist er an der Ostfalia in einem zweiten Forschungsprojekt zum Thema Codegenerierung tätig. Seine Schwerpunkte liegen in den Bereichen

AUTOSAR, modellbasierte Softwareentwicklung, Codegeneratoren und Eclipse-Technologien.



Kontakt

Internet: www.ostfalia.de
Email: mat.langer@ostfalia.de
Telefon: +49 5331 939 32410