

Modellbasierte Entwicklung von Autosar-Systemen

Die Zunahme der Komplexität von Software und Systemen im Auto und die daraus resultierenden Risiken bezogen auf Budgetüberschreitung, Lieferverzögerung und Qualitätseinbußen haben zur Gründung der Autosar-Initiative geführt. Seit kurzem gibt es auch eine Umgebung für **UML/SysML BASIERTES DESIGN** für die Entwicklung von Autosar-konformen Systemen und Komponenten.

Automotive Open System Architecture soll das Austauschen und Aktualisieren von Software und Steuergeräten während der Lebensdauer von Fahrzeugen erleichtern. Um dies zu erreichen, versteht Autosar ein System als eine Menge von Autosar Softwarekomponenten (SW-Cs), die miteinander über einen Autosar Virtual Functional Bus (VFB) kommunizieren und interagieren. Die SW-Cs werden auf spezifische Steuergeräte (ECUs, Electronic Control Units) abgebildet, die über das Netzwerk verteilt sind. Dank einer in Schichten angelegten Architektur können die SW-Cs auf andere Plattformen übertragen werden, ohne dass die funktionale Integration beeinträchtigt wird. Bewährte Autosar-konforme Komponenten sind so ohne langwierige und fehleranfällige Anpassungen auf eine andere Zielumgebung zu portieren.

Der Autosar VFB stellt eine Art Kommunikationsprotokoll für den Austausch von Informationen zwischen den SW-Cs dar. Die Autosar RTE (Run Time Environment) ist die Laufzeitumgebung, in der dieses Kommunikationsprotokoll implementiert ist. Bild 1 zeigt eine Darstellung der wichtigsten Autosar Elemente. Der Prozess ist so angelegt, dass der gesamte Weg der Softwarespezifikation abgedeckt ist, von der höchsten Abstraktionsebene bis hinunter zu den Details, wie einzelne Codefragmente einer Softwarekomponente innerhalb einer ECU ablaufen sollen. Eine SW-C kann ein ganzes System, ein Subsystem oder ein einzelnes, nicht weiter zerlegbares Funktionsmodul repräsentieren. Auf dieser untersten Ebene ist die SW-C auf einem einzelnen Steuergerät zu implementieren. Der Autosar VFB ist so angelegt, dass die Kommunikation von der ausführenden Technologie abstrahiert wird, so dass es keine Rolle spielt, ob zwei SW-Cs, die miteinander kommunizieren, auf verschiedenen über den Bus verbundenen ECUs oder auf derselben ECU implementiert sind.

Beschreibung eines Autosar-Systems mit Hilfe der UML/SysML

Graphische Modelle sind ein mächtiges Werkzeug für die anschauliche und gleichzeitig präzise Beschreibung komplexer Systeme. Sie abstrahieren den intelligenten Entwurf von der Implementierung und reduzieren gleichzeitig das Risiko von Missverständnissen bei der Kommunikation. Die Kombination aus UML (Unified Modeling Language) und SysML (Systems Modeling Language), ergänzt um ein Autosar spezifisches Profil unterstützt speziell das Design und die Spezifikation von Autosar-konformen Systemen mit Hilfe der folgenden Diagramme:

- das Systems Diagramm zur Erfassung des gesamten Autosar-Systems,
- das Software Component Diagramm zur Festlegung der Software-Architektur,
- das Internal Behavior Diagramm zur Spezifizierung der Schnittstelle zwischen dem Autosar-System und dem standardisierten Autosar Run-Time Environment (RTE), in dem das System laufen soll (Bild 1),
- das ECU Diagramm zur Festlegung der Typen von Steuergeräten (Electronic Control Units, ECU) und ihrer Kommunikationsschnittstellen,
- das Topology Diagramm zur Festlegung der physikalischen Topologie oder der physikalischen Architektur des Systems einschließlich aller ECUs im Fahrzeug und ihrer Verschaltung.

Die funktionale Perspektive

In der Regel beginnt die modellbasierte Herangehensweise mit der Darstellung der funktionalen Perspektive des zu entwickelnden Systems zusammen mit der Kommunikation über den Autosar VFB. Dies geschieht in Form eines Software Component Diagramms. Die Kommunikation wird definiert mit Hilfe spezifischer Ableitungen der bekannten UML Ports: Server Port, Client Port, Sender Port und Receiver Port. Ein Server Port liefert Daten, die ein Client Port angefordert hat,

und ein Sender Port liefert Daten, die ein Receiver Port verarbeitet.

Für das Subsystem Fensterheber mit dem Namen PowerWindow werden anschließend die atomaren Autosar SW-Cs definiert. Dies sind Funktionsgruppen, die auf demselben Steuergerät laufen. In diesem Beispiel sind dies: ein MainController, auf dem die ganze Kontroll-Logik zentralisiert wird und vier Instanzen vom Typ WindowController, für jedes Fenster eine:

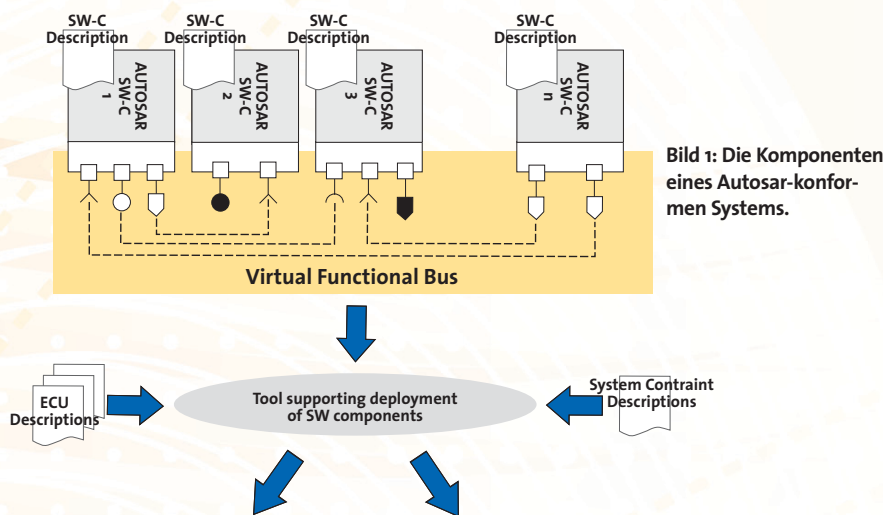
- FRWnd:WindowController,
- RRWnd:WindowController,
- RLWnd:WindowController und
- FLWnd:WindowController.

Im nächsten Schritt werden die Schnittstellen zwischen diesen Autosar SW-Cs spezifiziert. Die benötigte Bandbreite soll so gering wie nötig sein, und der Informationsfluss läuft ausschließlich vom MainController zu den vier WindowControllers. Dazu erhält der Hauptprozessor vier Sender Ports, und jeder WindowController einen Receiver Port. Eine Verbindungslinie beschreibt, welcher Sender Port Informationen an welchen Receiver Port sendet. Darüber hinaus wird nun auch festgelegt, welche Datenelemente auf jedem Kommunikationskanal übermittelt werden.

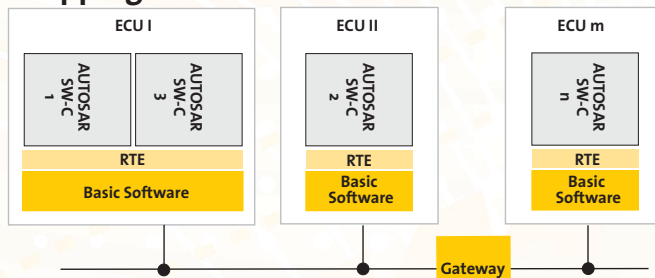
Die Innenansicht einer Autosar-Komponente

Mit Hilfe des Internal Behavior Diagramms wird nun der Ablauf der einzelnen ausführbaren Elemente innerhalb einer jeden Autosar SW-C spezifiziert. Ein ausführbares Element ist ein Codefragment, ein Algorithmus, eine Funktion oder eine Kombination aus diesen Elementen, die unabhängig von anderen Elementen ausgeführt werden kann. Das Signal zur Ausführung eines solchen Elementes wird von einem Triggerereignis ausgelöst, z.B. ein ablaufender Timer oder eine Messgröße, die einen Schwellwert überschreitet. Diese Rahmenbedingungen und Ausführungsregeln werden im Internal Behavior Diagramm definiert.

VFB view



Mapping



niert, jedoch ohne die tatsächliche Implementierung der einzelnen ausführbaren Elemente festzulegen.

Unter dem Namen WindowCtrl_IB werden die ausführbaren Elemente beschrieben. In diesem Fall gibt es ein ausführbares Element mit dem Namen R1. Nun werden die Ereignisse definiert, welche die Ausführung von R1 auslösen, sowie die Daten, welche über den Autosar VFB bzw. die Laufzeitumgebung übermittelt werden. Die elektrische Architektur des umgebenden Fahrzeugs wird im Autosar Prozess ebenfalls beschrieben. Dazu bedient man sich einer Kombination von ECU Diagramm und Topology Diagramm. Diese basieren auf den UML/SysML Klassen- und Objektdiagrammen und erlauben die Beschreibung der Steuergeräte, ihrer Ports sowie der CAN, LIN, FlexRay oder MOST Bussysteme, über welche die Steuergeräte miteinander verbunden sind.

Vom Modell auf das Steuergerät

Der letzte Schritt bei der Erstellung eines Autosar-System Modells ist die Festlegung, wie die Autosar Softwarekomponenten auf die Steuergeräte abgebildet werden. Im System Diagramm wird für jede Autosar SW-C, die im Software Component Diagramm beschrieben wurde, festgelegt, auf welchem Steuergerät im umgebenden Fahrzeug (wie im ECU Diagramm definiert) sie implementiert werden soll. Nachdem diese Zuordnung beschrieben ist, kann die Kommunikationsmatrix hergeleitet werden. In unserem Beispiel wird der MainController der

MainECU zugeordnet, und die Softwarekomponenten für die einzelnen Fensterheber den Steuergeräten in den jeweiligen Türen.

Zusammenfassung

Für die Zuverlässigkeit, Funktionalität und Vermarktungsfähigkeit heutiger Automobile sind Software und Elektronik in hohem Maße ausschlaggebend. Da andererseits der Kostendruck steigt und die Entwicklungszyklen kürzer werden, stellt die Standardisierung und gemeinsame Verwendung von Basisfunktionen und Subsystemen eine viel versprechende Option dar, die Entwicklungsaufwendungen und die Risiken zu reduzieren. Mit einem UML/SysML basierten Ansatz, wie in Rhapsody Autosar Pack, stehen den Entwicklern alle Möglichkeiten zur Verfügung die Vorteile von Autosar und den Nutzen einer modellbasierten Entwicklung – bis hin zur automatischen Codegenerierung – in vollem Umfang zu nutzen. (tk) ←



*Rick Boldt, Sr. Director
Automotive Industry Marketing,
Telelogic.*



*Renate Stücka, Sr. Director
Business Development Central
Europe, Telelogic.*

infoDIRECT www.all-electronics.de

Link zu Telelogic:

610AEL050