

Architekturmodellierung mit **EAST-ADL2** und **AUTOSAR**

TOP-DOWN-DESIGN VON
VERTEILTEN EMBEDDED SYSTEMEN

Um Timinganforderungen berücksichtigen zu können, werden mit der Einführung von AUTOSAR 4.0 bis zum Ende des Jahres auch Timing Constraints unterstützt. Trotz des mittlerweile sehr mächtigen Standards können nicht alle Aspekte und Anforderungen an das EE-Architekturdesign adressiert werden. Dieser Problematik haben sich mittlerweile alternative Standards wie EAST-ADL2 und die Timing Extension TIMMO angenommen.

Die Absicherung von sicherheitskritischen Systemen im Automotive-Umfeld kann ohne Berücksichtigung des Timings nicht ausreichend gewährleistet werden. Verletzungen von Timing Constraints können zu Fehlfunktionen und im „Worst Case“ zu Fahrzeug- und Personenschäden führen.

Um Timinganforderungen berücksichtigen zu können, werden mit der Einführung von AUTOSAR 4.0 bis zum Ende des Jahres 2009 auch Timing Constraints unterstützt. Durch die Kombination von AUTOSAR mit East-ADL2 und der TIMMO Timing Language TADL wird ein durchgängiger Top-Down-Designansatz unterstützt, sowohl auf funktionaler als auch auf der Timing-Ebene.

Ursprünglich wurde EAST-ADL (Electronic Architecture and Software Tools, Architecture Description Language) innerhalb des EAST-EEA European Project entwickelt. Innerhalb des ATTEST-Projekts wurde ein weiteres Release EAST-ADL2 erarbeitet, welches Vorschläge zur Integration von AUTOSAR beinhaltet. EAST-ADL2 definiert verschiedene aufeinander aufbauende Abstraktionsebenen (**Bild 1**).

Feature (Vehicle)-Ebene

Auf der obersten Ebene stehen vor allem die Handhabung der verschiedenen Varianten und die Modellierung der Kundenfunktionen (Features) im Vordergrund. So soll erkennbar werden, welche verschiedenen Funktionalitäten

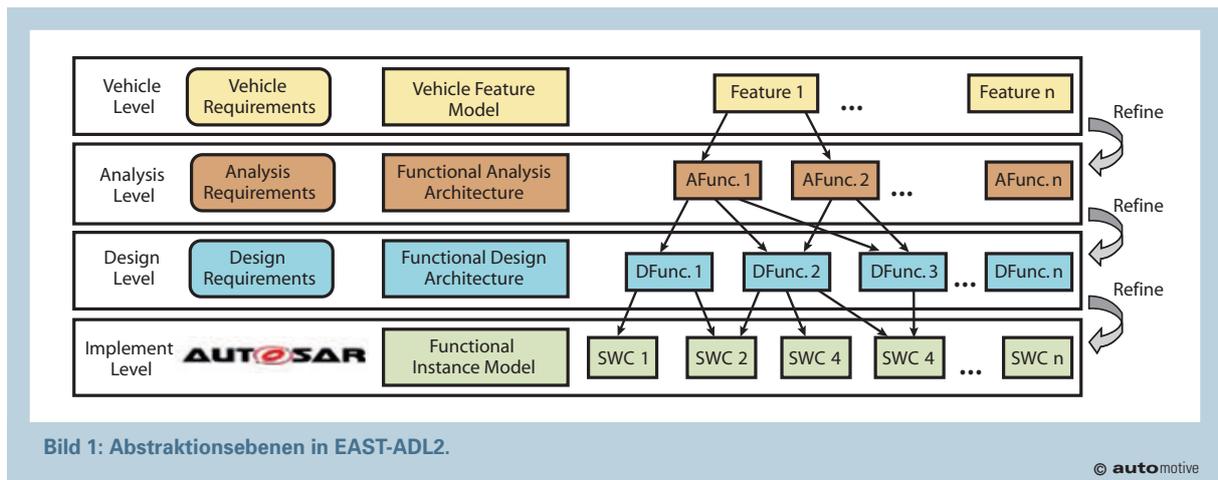
für eine bestimmte Variante des Systems vorhanden sein müssen. Mittels dieses Artefakts wird somit eine Unterstützung für eine Produktlinie angeboten. Der Aufbau des Vehicle View ist so gewählt worden, dass die Eigenschaften von einer anderen Variante übernommen werden können. Beispiele für Features sind Sitzheizung, Adaptive Cruise Control oder Scheibenwischer. Varianten wären beispielsweise Scheibenwischer mit oder ohne Regensensor.

Analyse-Ebene

Die auf der Feature-Ebene definierten Funktionalitäten und deren Abhängigkeiten untereinander werden auf der Analyse-Ebene detailliert. Dadurch ergibt sich eine n-zu-m-Beziehung zwischen dem Vehicle View und der Analyse-Architektur. Anhand dieser Beschreibung erfolgt eine strukturelle Analyse der resultierenden Funktionalen Netzwerke. Funktionsduplikate können identifiziert und optimiert werden. Ebenso wird das funktionale Verhalten abgebildet, welches über Simulation verifiziert werden kann.

Design-Ebene

Die Design-Architektur ist wiederum eine Verfeinerung der rein funktionalen Analysearchitektur. Sie beschreibt schließlich die HW/SW-Partitionierung auf Basis der Hardware-Architektur.



Implementierungsebene

Abgeleitet von der Design-Architektur entsteht die Implementierungsebene. Diese Ebene repräsentiert die "flache" SW-Architektur, welche auf die AUTOSAR-Abstraktionsebene gemapped werden kann. Abhängig vom Detaillierungsgrad der Implementierungsebene kann das Mapping auf AUTOSAR-Softwarekomponenten-Ebene oder Runnable-Ebene erfolgen.

Auf sämtlichen Ebenen werden Requirements (sowohl funktionale als auch nichtfunktionale Requirements wie Timingconstraints) definiert, die von Ebene zu Ebene ebenfalls detailliert werden.

Beschreibung und Abbildung von Timing-Aspekten

Im Rahmen des ITEA-Projekts TIMMO (TIMing MOdel) wurde EAST-ADL2 um ein Timing-Modell erweitert, welches in die relevanten Abstraktionsebenen integriert ist. Die verschiedenen in EAST-ADL2 definierten funktionalen Ebenen nutzen sogenannte ADLFunctionTypes bzw. ADLFunctionPrototypes, welche über ADLFlowPorts Informationen mit anderen Funktionen austauschen. Für Input-Ports können Events mit unterschiedlichen Trigger-Policies definiert werden. Anhand der funktionalen Netzwerkbeschreibungen der unterschiedlichen in EAST-ADL2 definierten Abstraktionsebenen resultieren daraus Signal-Ketten (Event Chains) mit unterschiedlicher Granularität. Basierend auf den Signalketten können mit Hilfe der in TIMMO definierten Timing-Beschreibungssprache TADL (Timing Augment Description Language) Timing Constraints für beliebige Segmente definiert werden. Folgende Timing Constraints-Konzepte werden unterstützt:

■ End-to-End Delay (Age)

Ein Delay Requirement definiert eine maximal zulässige Verzögerungszeit von Daten zwischen einem Sender und einem Receiver. So muss z. B. ein aktuelles Bremssignal nach einer Mindestzeit einer Objekterfassung mittels Radar eines ACC-Systems unabdingbar vorhanden sein.

■ Synchronization Constraints

Signale müssen in gewissen Anwendungsfällen synchron zueinander sein. Beispielsweise müssen die von den vier

Rädern kommenden Geschwindigkeitssignale, um sie sinnvoll vergleichen zu können, am Bremsregler-Eingang vom gleichen Datum sein. Ebenso müssen die Bremsaktuatorsignale, um ein „korrektes“ Bremsen zu gewährleisten, synchron zueinander sein.

■ Event Constraints

Je nach Anwendung müssen Events periodischer, sporadischer Natur sein oder einem vordefinierten Muster folgen. Die Repetition Rate (Periodic Event) definiert die Empfangszeit von Daten an einem Port oder die Triggering Periode von ADLFunctions. Das ist beispielsweise die Min-

dest-Abtast-Häufigkeit eines Sensor-Signals, damit dieses zuverlässig erfasst werden kann.

Die Beschreibung von Timing Aspekten in TIMMO soll am Beispiel ACC (Adaptive Cruise Control) näher erläutert werden. Das Grundprinzip eines ACC-Systems funktioniert folgendermaßen: Die in **Bild 2** gezeigte Darstellung zeigt einen Ausschnitt aus einem ACC-System auf der AUTOSAR-RTE-Ebene, die auf die EAST-ADL2-Implementierungsebene gemapped wird. Das durch den Sensor „Radar“ erzeugte Signal durchläuft die unterschiedlichen Hard- und Software-Schichten. Bei jedem Übergang in eine neue Schicht wird ein Event erzeugt. Zwei benachbarte Events bilden dabei ein Timing-Segment. Durch die Verzweigung der Signalpfade kommt es zu Baumstruktur-artigen Event-Ketten. Diesen können dann sinnvolle Timing Constraints zugewiesen werden. Relevante Timing Constraints für das Beispiel ACC könnten das End-to-End Timing Delay zwischen Objekterfassung am Radar und Bremsaktuator sein, oder die Synchronität der an den Rädern erfassten Geschwindigkeitssignale.

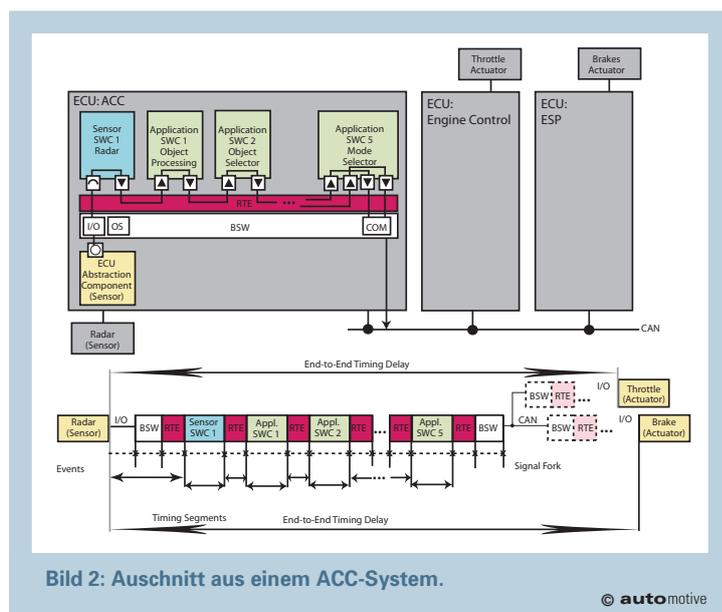


Bild 2: Ausschnitt aus einem ACC-System.

© automotive

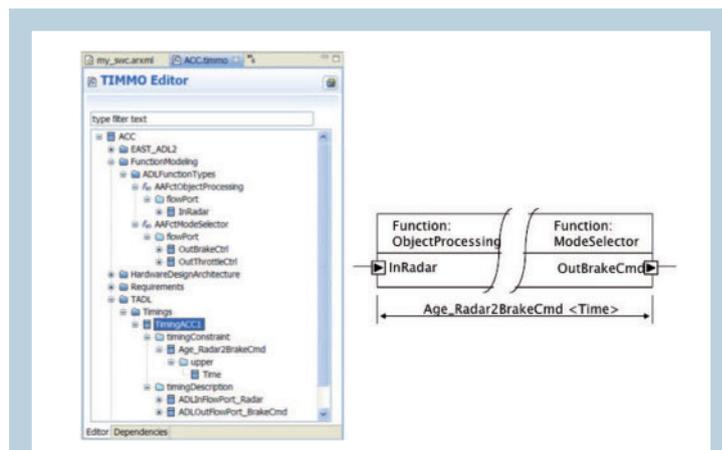


Bild 3: Definition eines End-to-End-Timing Constraint im VSA TIMMO-Editor.

© automotive

Netzwerkdesigntool VNA

Erst mit der Existenz eines parametrisierbaren Timing-Modells können sowohl Timing Requirements in einem frühen Entwicklungsstadium erfasst, als auch das Timing-Verhalten von verteilten Embedded Systemen analysiert und optimiert werden. Diese Prinzipien sind beispielsweise in dem Netzwerkdesigntool VNA (Vehicle Network Architect) umgesetzt, welches zwei Betriebs-Modi unterstützt:

■ Analysemodus

Basierend auf einer bestehenden Kommunikationsmatrix können mit Hilfe der Deadline Monotonic Analysis (DMA) Worst-Case-Signallaufzeiten und -Buslast für CAN- und/oder LIN-Netzwerke berechnet werden.

■ Synthesemodus

In diesem Modus werden als Input Timing-Anforderungen in Form von maximal zulässigen Signal-Latenzzeiten und die maximal zulässige Buslast vorgegeben. Anhand dieser Informationen wird über einen Optimierungsalgorithmus das Signal-zu-PDU und PDU-zu-Frame-Packing automatisch generiert. Scheduling-Tables für Gateways werden ebenfalls automatisch erstellt. Dies führt zu einer gültigen Netzwerkkonfiguration unter Einhaltung der definierten Requirements.

Toolunterstützung mit Volcano VSA

Seit Anfang dieses Jahres ist das Architekturwerkzeug VSA (Volcano Vehicle Systems Architect) auf dem Markt verfügbar. Basierend auf der Eclipse-Technologie unterstützt es sowohl das AUTOSAR- als auch das TIMMO/EAST-ADL2-Metamodell. Somit lässt sich eine logische Fahrzeugarchitektur einschließlich der Timing-Aspekte nahtlos auf allen nötigen Abstraktionsebenen beschreiben. Unterstützt von „built-in“ oder benutzerdefinierten Constraints kann das Design in jedem Entwicklungsschritt auf seine Richtigkeit und Vollständigkeit hin verifiziert werden. Bereits vorhandene domainspezifische grafische Editoren unterstützen auch den „gelegentlichen“ Anwender in den unterschiedlichen Stufen der Architekturentwicklung und in der AUTOSAR-Softwarekonfiguration. Durch die Berechnung von Metriken wie Performance, Ressourcen-Auslastung oder Kosten soll in Zukunft auch die Evaluierung von Architekturvarianten machbar sein. Dank der bereits vorhandenen Schnittstelle zu Kabelbaum-Entwicklungswerzeugen wie CHS wird die Brücke zum physikalischen Architekturdesign geschlagen.

Fazit

Basierend auf dem TIMMO-Metamodell können E/E-Architekturen auf unterschiedlichen Abstraktionsebenen bis hin zur Implementie-

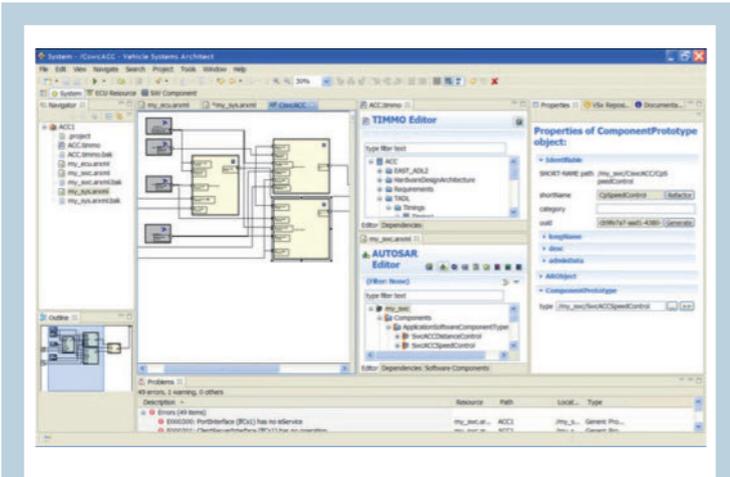


Bild 4: VSA – Volcano Vehicle Systems Architect.

© automotive

Daher können Zeit und Geld anhand eines Frontloaded Designprozesses in zweierlei Hinsicht gespart werden. Kostenaufwendige Hardwaretests können reduziert und Qualität und Zuverlässigkeit des Endprodukts „Fahrzeug“ erhöht werden. (oe)

Literatur

- [1] <http://www.timmo.org>
- [2] <http://www.autosar.org>
- [3] An Alternative View on Model-based Development for Automotive Embedded Systems, Daniel Karlsson, Volvo Technology Corporation, Göteborg, Sweden
- [4] The TIMMO Methodology, TIMMO Open Workshop, 26. März 2009, Stefan Kuntz, Continental Automotive GmbH

rungsebene hinsichtlich Funktion und Timing beschrieben werden. Dadurch ist auch ein nahtloser Übergang zur AUTOSAR-Methodik und somit zur Softwarekonfiguration gegeben. Sowohl Top-Down- als auch Bottom-Up-Design werden auf diesem Wege unterstützt.

Eine vollständige Funktionsverifikation von Systemen und Subsystemen kann letztendlich nur unter Berücksichtigung der Timing-Aspekte erfolgen. Diese entscheiden oftmals über Funktionieren oder Nichtfunktionieren eines Systems.



Michael Seibt ist bei Mentor Graphics im Bereich Automotive Network Design als Produktmanager tätig.

@ Mentor Graphics
www.mentor.com