

# AUTOSAR

## im Entwicklungsprozess

Vorgehen bei der Serien-Einführung der modellbasierten



# ess

## AUTOSAR-Funktionsentwicklung



Der AUTOSAR-Standard definiert eine Softwarearchitektur, Software-schnittstellen und Austauschformate, die den Aufbau einer steuergereäteherstellerunabhängigen Funktionsbibliothek ermöglichen. Bei der Daimler AG wird deshalb im Bereich Komfort- und Innenraumfunktionen der AUTOSAR-Standard im Rahmen der Standardisierung von Softwaremodulen der Funktionsbibliothek eingeführt. Dabei kommt der Serien-code-Generator Target-Link zur Modellierung und Code-Generierung für AUTOSAR-Softwarekomponenten zum Einsatz.



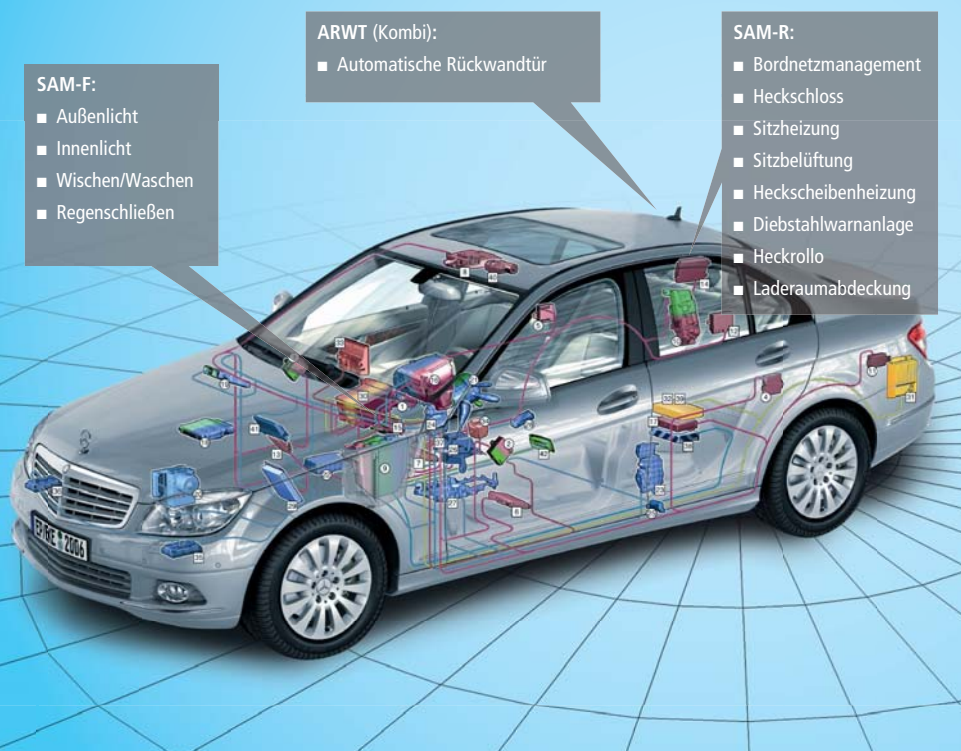


Bild 1: Modellbasiert erstellte Komfortfunktionen der C-Klasse (Baureihe 204), für die automatisch erzeugter Code eingesetzt wurde (SAM: Zentrale Steuergeräte, ARWT: Automatische Rückwandtür).

Die modellbasierte Entwicklung und automatische Seriencode-Generierung wird bei der Daimler AG in der Pkw-Entwicklung eingesetzt. Der Beitrag beschreibt das Vorgehen bei der schrittweisen Umstellung der modellbasierten Entwicklung auf die AUTOSAR-Architektur und berichtet von den bisherigen Erfahrungen.

### Modellbasierte Entwicklung im Innenraumbereich

Fahrzeugfunktionen und speziell Komfort- und Innenraumfunktionen werden seit einigen Jahren bei der Daimler AG modellbasiert entwickelt. Schwerpunkt der modellbasierten Entwicklung im Innenraum sind Hauptfunktionen der Zentralsteuergeräte (Central Body Computers), in denen jeweils mehrere Funktionen integriert sind. In größerem Umfang wurde die modellbasierte Entwicklung im Innenraum der aktuellen C-Klasse (Baureihe 204) eingeführt (Bild 1). In den zurzeit in der Serienentwicklung befindlichen Baureihen wird der modellbasierte Ansatz

konsequent weiterverfolgt, dabei nimmt die Anzahl der zu integrierenden Funktionen stetig zu. Die von uns entwickelten Funktionen umfassen entweder die gesamte Steuergeräte-Applikationssoftware oder stellen sogenannte Mehrwertfunktionen dar, mit denen vorhandene Steuergeräte durch wettbewerbsdifferenzierende Funktionen ergänzt werden. Je Entwicklungszyklus erhält der Lieferant das von uns entwickelte Funktionsmodell sowie eine Testspezifikation und ist für die Umsetzung des Modells in Software sowie die Implementierung auf dem Steuergerät verantwortlich (Bild 2). Der Fahrzeughersteller hat durch die modellbasierte Entwicklung im Wesentlichen folgende Vorteile:

- Frühzeitige Reifegradabsicherung durch Simulation
- Darstellung und Absicherung von Funktionen im Fahrzeug bereits vor Verfügbarkeit der Steuergeräte durch Rapid-Prototyping

### Innenraum – CAN

1. Elektronisches Zündschloss (EZS)
2. Türsteuergerät vorne links (TSG VL)
3. Türsteuergerät vorne rechts (TSG VR)
4. Türsteuergerät hinten links (TSG HL)
5. Türsteuergerät hinten rechts (TSG HR)
6. Sitzverstellung Fahrer (SSG-F)
7. Sitzverstellung Beifahrer (SSG-BF)
8. Dachbedieneinheit (DBE/SHD)
9. SAM/SRB vorne
10. SAM/SRB hinten
11. Reifendruckkontrolle (RDK) (USA)
12. Anhänger Anschlussgerät (AAG)
13. Parktronic-System (PTS)
14. Keyless Go (KG)
15. Klimatisierungsautomatik (KLA)
16. Kombiinstrument (KI)
17. Multifunktionssteuergerät (MSS)
30. COMAND
37. Weight Sensing System (WSS) (USA)
40. Panorama-Schiebedach

### Fahrwerk – CAN

1. Elektronisches Zündschloss (EZS)
9. SAM/SRB vorne
16. Kombiinstrument (KI)
18. Motorelektronik (Benziner)
19. Motorelektronik (Diesel)
20. Electronic Stability Program (ESP)
21. Mantelrohrmodul (MRM)
22. Airbag ARAMIS
23. Reversibler Gurtstraffer vorne links
24. Reversibler Gurtstraffer vorne rechts
41. Adaptives Dämpfungs-System (ADS)

### Fahrdynamik – CAN

20. Electronic Stability Program (ESP)
25. Sensorcluster

### Antrieb – CAN

18. Motorelektronik (Benziner)
19. Motorelektronik (Diesel)

26. Elektronisches Wählhebelmodul (EWM)
27. Vollintegrierte Getr. Steuerung (VGS)
28. Tanksteuergerät (Benziner)
29. Elektronische Getr. Steuerung (EGS)

### LIN – Bus

2. Türsteuergerät vorne links (TSG VL)
3. Türsteuergerät vorne rechts (TSG VR)
9. SAM/SRB vorne
10. SAM/SRB hinten
18. Motorelektronik (Benziner)
19. Motorelektronik (Diesel)
21. Mantelrohrmodul (MRM)

### Telematik – CAN

30. COMAND
33. Zentraldisplay
34. COMAND Controller

### Private – Bus

15. Klimatisierungsautomatik (KLA)
42. Klimatisierungsautomatik Fond (KLA)

### MOST – Ring

30. COMAND
31. Soundverstärker
32. Digital Radio (DAB) (ECE)
39. SIRIUS Satellite Radio (USA)

### Diagnose – CAN

9. SAM/SRB vorne
38. Notrufsystem (USA)

### Frontbereich – CAN

9. SAM/SRB vorne
35. Steuergerät Xenon Scheinwerfer Master
36. Steuergerät Xenon Scheinwerfer Slave

- Direkte Implementierung durch automatische Seriencode-Generierung
- Lieferantenunabhängige Funktionsentwicklung und Weiterentwicklung
- Wiederverwendung von Funktionen und Schutz geistigen Eigentums

### Einbettung in die Softwarearchitektur

Die Integration der modellbasiert erstellten Funktionen in das Steuergerät durch den Lieferanten erfolgt bisher noch mit einem hohen manuellen Aufwand. Dieser ist sehr stark von der jeweiligen Softwarearchitektur des Lieferanten abhängig, auch wenn der Kommunikationsteil der Basissoftware von dem Fahrzeughersteller vorgegeben wird. Gegebenenfalls muss die Softwarearchitektur angepasst oder speziell erweitert werden. Eine bisher fehlende vollständig standardisierte Softwarearchitektur führt teilweise zu umfangreichen Abstimmungs-

gesprächen mit den jeweiligen Lieferanten. Der Abstimmungsbedarf geht jedoch über die Softwarearchitektur hinaus. So muss auch die Beschreibung der Metadaten zur Integration der Funktionen, zum Beispiel Schnittstellenlisten der Funktionen oder die Abbildung von Applikationssignalen auf Bussignale, gemeinsam zwischen Fahrzeughersteller und Steuergeräteleieferant definiert werden. Die Voraussetzung für einen prozesssicheren und breiteren Einsatz der modellbasierten Entwicklung ist somit eine einheitliche lieferantenübergreifende Softwarearchitektur und eine standardisierte Beschreibung der Metadaten.

#### Die AUTOSAR-Softwarearchitektur

Der AUTOSAR-Standard definiert eine Softwarearchitektur für Steuergeräte, ein Integrationsverfahren und die dazu notwendigen Austauschformate. AUTOSAR adressiert

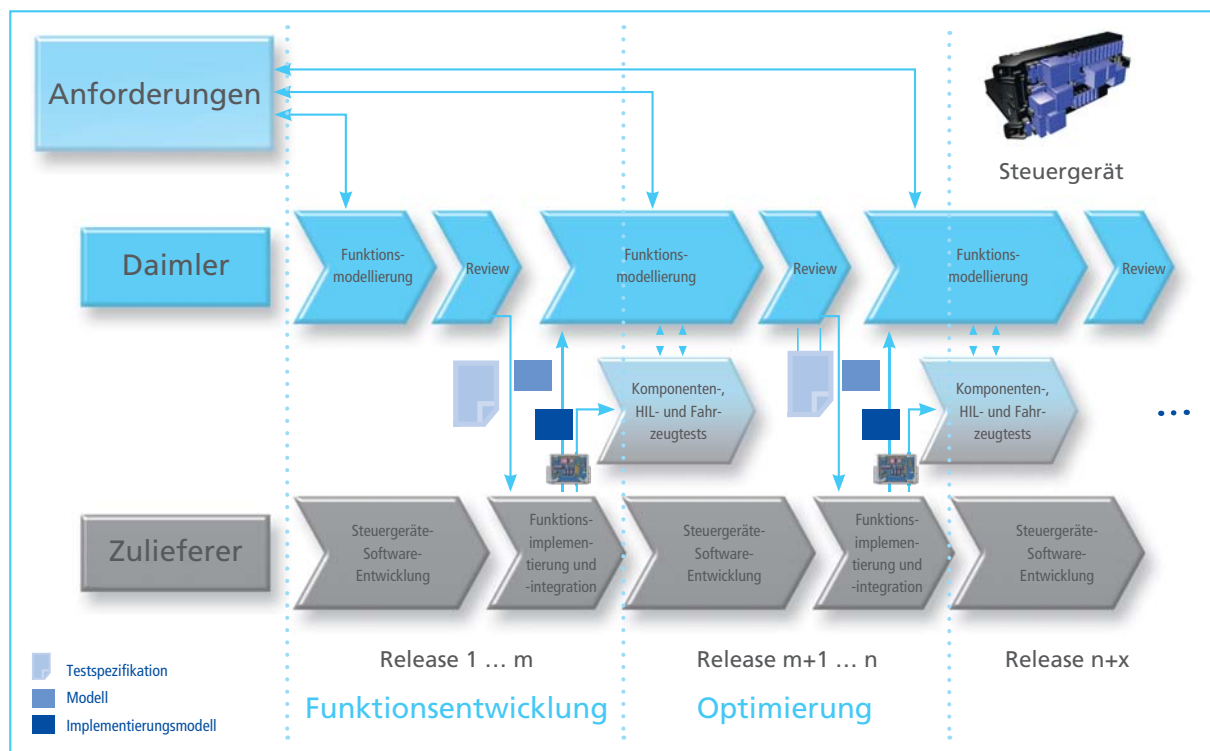
damit weitgehend die oben genannten Anforderungen an eine prozesssichere Integration von modellbasierten Funktionen. AUTOSAR gliedert die Applikationssoftware eines Steuergeräts in mehrere Softwarekomponenten (Software Components, SWC), die mittels einer Middleware (Run-Time Environment, RTE) miteinander kommunizieren. SWCs kapseln und typisieren die Software und ermöglichen einen Datenaustausch nur über wohldefinierte Schnittstellen.

Für die Integration auf ein Steuergerät sind zwei Mapping-Schritte erforderlich: erstens ein Mapping der SWC-Instanzen auf Steuergeräte und zweitens im Fall von nicht-lokaler, also steuergeräteübergreifender Kommunikation ein Mapping der Datenelemente auf Netzwerksignale.

#### AUTOSAR-Einführungsstrategie

Für die nächste Generation von Innenraumsteuergeräten der Daimler AG wird ein erster Schritt in Richtung AUTOSAR-Architektur angegangen. Die Einführung erfolgt hier im Schichtenmodell der AUTOSAR-Softwarearchitektur von oben nach unten, d. h. angefangen bei den Applikations-Softwarekomponenten (SWCs) und der RTE (Bild 3). Dazu wird die Softwarearchitektur konsequent in Applikations- und Basissoftwareanteile aufgeteilt, die über eine definierte Schnittstelle miteinander kommunizieren. Basis für die Definition dieser Schnittstelle ist der AUTOSAR-Standard, der hier jedoch teilweise eingeschränkt und an einigen Stellen präzisiert wird. Die Standard-Softwarearchitektur basiert in diesem Schritt weiterhin auf dem etablierten Daimler-Standard-Core und wird um ausgewählte AUTOSAR-Softwaredienste (wie Memory Management von NVRAM-

Bild 2: Typischer Entwicklungszyklus mit mehreren Iterationen.



Daten) ergänzt. In diesem ersten Schritt ist die Netzwerk-Kompatibilität der so entwickelten Steuergeräte mit klassisch entwickelten Steuergeräten weiterhin gegeben. Damit ist eine schrittweise Einführung der AUTOSAR-Technologie möglich.

### Modellierung auf Verhaltens- und Architekturebene

Mit AUTOSAR geht eine Modellierung auf zwei Ebenen einher. Neben der Modellierung des Verhaltens der Funktionen ist nun zusätzlich eine formale Beschreibung der Schnittstellen der Funktionen in Form von SWCs und deren Vernetzung erforderlich (Architekturebene). Für neu zu erstellende Fahrzeugfunktionen empfiehlt sich im Sinne einer Top-down-Strategie, zuerst die Zergliederung der Funktionalität in mehrere SWCs und die Definition ihrer Schnittstellen auf Architekturebene vorzunehmen (Bild 4). Anschließend wird das Verhalten der so zergliederten SWCs modelliert. Bei bereits vorhandenen Funktionsmodellen kann bottom-up aus den Modellschnittstellen die SWC-Beschreibung erzeugt werden.

Die entstandenen SWCs werden anschließend auf Architekturebene untereinander verbunden. Bei einer schrittweisen Einführung ist ein vollständiger Top-down-Entwurf eines gesamten Fahrzeugs nicht möglich. Umgekehrt liegen für ein Steuergerät nicht alle Funktionen von Anfang an als Modell vor. Daher hat sich eine „Meet-in-the-Middle-Strategie“ bewährt, die für ein Steuergerät

- die SWC-Beschreibung aus existierenden Verhaltensmodellen ableitet,
- bei neuen Fahrzeugfunktionen zunächst die Schnittstellen auf Architekturebene über SWCs definiert und
- Sensor-/Aktor-SWCs mittels fester Regeln automatisch erzeugt.

Die so entstandenen SWCs werden anschließend auf Architekturebene in einer Composition zusammengefasst und untereinander vernetzt, die verbliebenen unverbundenen Ports werden nach außen geführt und damit zu Ports der Composition. Diese Ports repräsentieren nun die

Kommunikationsschnittstelle des Steuergeräts und die durch die Ports referenzierten Datenelemente können auf die durch die Kommunikationsmatrix vereinbarten Signale des Steuergeräts abgebildet (gemappt) werden. So lässt sich die SWC-Struktur eines Steuergeräts mit vertretbarem Aufwand erstellen. In den folgenden Abschnitten werden diese Tätigkeiten genauer erläutert.

### Festlegen der AUTOSAR-Interfaces

Basis für die Entwicklung von Softwarekomponenten ist eine gemeinsame Datenbank (Common Object Pool, COP) von Interface- und Datentyp-Definitionen, die der SWC-Entwickler bei der Definition der SWC-Ports nutzen kann. Diese Datenbank wurde aus den Signaldefinitionen der Kommunikationsmatrix (K-Matrix) abgeleitet und muss mit dieser laufend abgeglichen werden (Bild 8).

Die Ableitungsregeln hierzu legen fest, dass für jedes K-Matrix-Signal ein Interface mit einem Datenelement erzeugt wird. Dadurch wird der Übergang von der bisherigen K-Matrix-Signal-Welt in die AUTOSAR-Welt erleichtert, da anhand der Namensgleichheit die passenden Interfaces leicht gefunden werden können. Die Strukturierungsmöglichkeiten von AUTOSAR, mehrere Datenelemente zu einem Interface zusammenzufassen, werden bei neuen Interfaces genutzt, die abgeleiteten Interfaces profitieren hiervon jedoch vorerst nicht. Um die Kompatibilität der SWCs untereinander zu gewährleisten, wird durch Modellierungsregeln vereinbart, dass lediglich Interfaces verwendet werden dürfen, die in der Datenbank vorhanden sind. Gegebenenfalls muss diese Datenbank durch einen Koordinator um zusätzliche Interfaces erweitert werden.

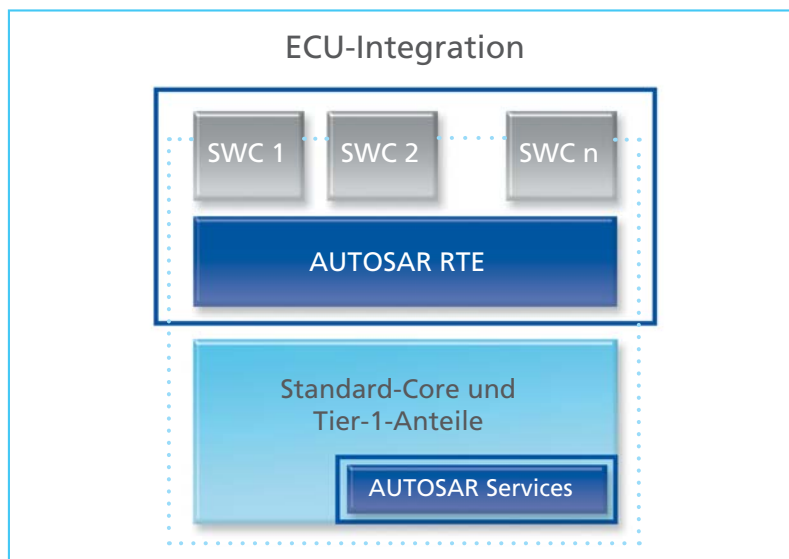


Bild 3: Bei der schrittweisen AUTOSAR-Einführung werden zunächst Applikationssoftware (SWCs) und Middleware (RTE) AUTOSAR-konform entwickelt. Die Basissoftware besteht weiterhin aus dem Daimler-Standard-Core, erweitert um ausgewählte AUTOSAR-Softwaredienste.

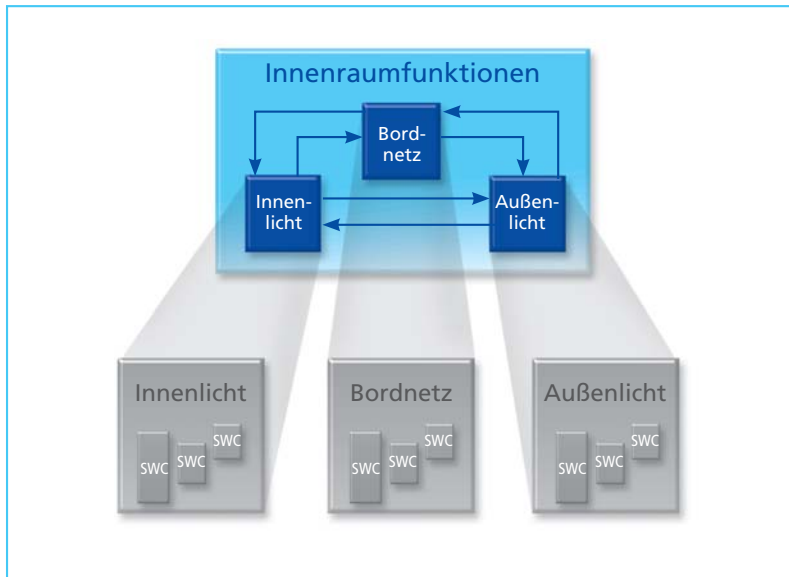


Bild 4: Beim Top-down-Ansatz erfolgt zunächst die Zergliederung der Funktionalität in mehrere SWCs und die Definition ihrer Schnittstellen auf Architekturebene.

### Erstellung neuer Fahrzeugfunktionen entsprechend der Top-down-Strategie

Die Modellierung neuer Fahrzeugfunktionen beginnt auf Architekturebene mit der Beschreibung der SWC-Schnittstellen in der oben genannten COP-Datenbank. Dazu werden Ports angelegt, die durch Interfaces typisiert werden. Weiterhin werden hier Runnables und deren Teilmenge der SWC-Schnittstelle vereinbart. Über RTE-Events werden die Zykluszeit der Runnables und gegebenenfalls deren ereignisgesteuerte Aktivierung, zum Beispiel beim Empfang eines neuen Signals, vereinbart. Anschließend wird daraus eine AUTOSAR-SWC-Beschreibung generiert.

Die Verhaltensmodellierung erfolgt auf Basis dieser AUTOSAR-Beschreibung. Dazu wird die SWC-Beschreibung in das Data Dictionary importiert und ein Modellierungsrahmen generiert, der einen Ausgangspunkt für die Modellierung der Verhaltensfunktion bietet. Dieser Modellierungsrahmen enthält die relevanten AUTOSAR-Schnittstellen und Runnables unter

Nutzung der TargetLink-AUTOSAR-Blöcke (Bild 5).

Der Rahmen wird anschließend mit Standard-Blöcken zur Modellierung der eigentlichen Algorithmik gefüllt. Änderungen an der Schnittstelle werden stets in der Datenbank vorgenommen.

### Migration existierender Modelle entsprechend der Bottom-up-Strategie

Bestehende Verhaltensmodelle wurden migriert. Dabei kam die vorhandene Schnittstellendefinition für die Ableitung der Ports und deren Typisierung zum Einsatz. Nahezu 90 % der Modellumfänge konnten hier automatisiert migriert werden, da bereits Modellierungskonventionen feststanden, bei Eingangs- und Ausgangsvariablen den Namen des zugehörigen K-Matrix-Signals zu verwenden. Zunächst wurde im Data Dictionary die SWC angelegt und im Modell wurden die entsprechenden AUTOSAR-Blöcke eingefügt (Bild 6). Anschließend wurde die bei der Code-Generierung entstehende XML-Datei (SWC-Beschreibungs-

## Glossar

**Composition** – Eine Composition bezeichnet die Zusammenfassung und Verschaltung mehrerer SWCs zu einer übergeordneten Komponente.

**DBC-Datei** – Datenbankcontainerdatei. Kommunikationsmatrix-Beschreibungsdatei für CAN.

**K-Matrix** – Die Kommunikationsmatrix beschreibt die zwischen einzelnen Steuergeräten ausgetauschten Signale/Informationen.

**NVRAM** – Non volatile Random Access Memory. Nicht flüchtiger Speicher wird beispielsweise zur Speicherung von Diagnoseinformationen verwendet.

**Runnable** – Die ausführbaren Elemente einer AUTOSAR-SWC, vergleichbar einer Funktion.

**Systemarchitekturwerkzeug** – Systemarchitekturwerkzeuge dienen der Planung und Integration komplexer Steuergeräte-Architekturen. Sie können AUTOSAR-Dateien importieren und exportieren sowie diese modifizieren.

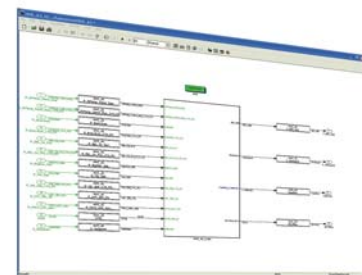


Bild 5: Ein mit TargetLink erstelltes AUTOSAR-Rahmenmodell.

datei) in die SWC-Datenbank eingelegt. Die weitere Pflege der Schnittstelle kann nun in der Datenbank erfolgen. Bei dieser Migration konnte die modellierte Funktionalität (Verhaltensmodell) vollständig und unverändert übernommen werden, lediglich die Modellschnittstelle musste an den Standard angepasst werden (Bild 7).

### AUTOSAR-Werkzeugkette

Für die Entwicklung einer AUTOSAR-konformen Fahrzeugfunktion wurde



eine bestehende Werkzeugkette für AUTOSAR erweitert (Bild 8). Ein wesentlicher Baustein der Entwicklungsumgebung ist die COP-Datenbank zur Ablage von AUTOSAR-Interfaces und SWC-Beschreibungen. Sie wurde als Erweiterung einer bestehenden Datenbankanwendung zur K-Matrix-Erstellung realisiert. Neben den AUTOSAR-Modellierungselementen können hierin auch Parameter und NVRAM-Größen beschrieben und mit Werten versehen werden. Durch die Nähe zur K-Matrix-Entwicklung ist es möglich, AUTOSAR-Interfaces elegant aus deren Datenbeständen zu generieren und zu aktualisieren. Weiterhin konnte das erprobte Freigabekonzept der K-Matrix-Entwicklung übernommen werden. Über die Austauschformate von AUTOSAR lassen sich die SWC-Beschreibungen im- und exportieren. Dadurch ist die Kopplung zum Data Dictionary und zu Systemarchitekturwerkzeugen gegeben. Die modellbasierte Entwicklung von AUTOSAR-konformen Funktionsmodellen wird durch Model-in-the-Loop (MIL)- und Software-in-the-Loop (SIL)-Tests unterstützt, die in die Entwicklungsumgebung integriert sind. Abschließend wird AUTOSAR-konformer Seriencode zusammen mit einer konsistenten AUTOSAR-SWC-Beschreibungsdatei generiert.

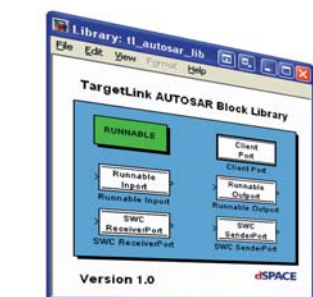


Bild 6: Die Funktionsmodellierung erfolgt unter Verwendung der AUTOSAR-Blöcke von TargetLink.

Die Systemarchitekturwerkzeuge werden zur Vernetzung der SWCs eingesetzt. Dies geschieht in der Regel durch grafische Editoren, die eine bessere Visualisierung als rein formularbasierte Datenbankanwendungen bieten.

Im Systemarchitekturwerkzeug wird heute die Abbildung der nicht-lokalen Datenelemente auf Netzwerksignale vorgenommen. Das dadurch entstehende AUTOSAR-System-Template wird zur Steuergeräte-Integration beim Lieferanten eingesetzt.

### Erkenntnisse und Ausblick

Die modellbasierte Funktionsentwicklung erfährt durch AUTOSAR eine seit Langem geforderte Standardisierung von Beschreibungsformaten und Schnittstellen. Die typischeren AUTOSAR-Beschreibungen ermöglichen es schon sehr früh im Entwicklungsprozess, die Konsistenz zwischen unabhängig entwickelten Funktionsmodellen sicherzustellen, wenn wir als Fahrzeughersteller die Funktionsmodelle dem Steuergeräteleveranten übergeben. Es ist deshalb zu erwarten, dass die Funktionsintegration beim Lieferanten sehr viel effizienter ablaufen wird. Abstimmungsrunden zwischen Fahrzeughersteller und Lieferanten zur Softwarearchitektur sind durch das mit AUTOSAR vereinheitlichte Vokabular wesentlich

produktiver. Der zurzeit notwendige Einsatz von zwei Entwicklungswerkzeugen (zur Verhaltensmodellierung und zur Schnittstellenbeschreibung) beim Entwickeln von AUTOSAR-konformen Funktionsnetzwerken schafft neue Herausforderungen bezüglich der Unterteilung eines Systems in handhabbare und logisch sinnvolle Softwarekomponenten. Der Übergang zwischen den verschiedenen Modellierungswerkzeugen oft unterschiedlicher Hersteller muss zukünftig im Sinne eines Roundtrips noch effizienter gestaltet werden. Weiterhin werden durch die aktuelle Aufteilung der AUTOSAR-Entwicklungsumgebung in Werkzeuge zur Architekturmodellierung und Systemintegration sowie zur Verhaltensmodellierung mit jeweils eigenständigen Code-Generatoren die realen Softwaresystemschnittstellen zu einem gewichtigen Teil durch die Werkzeugdomäne bestimmt. Eine flexible werkzeugübergreifende Systemmodellierung und Ressourcen-Optimierung stößt hier heute noch an ihre Grenzen. Architektur- und Designentscheidungen müssen daher a priori gut überlegt sein. Die Mächtigkeit der AUTOSAR-Beschreibung und die teilweise noch nicht vollständig vorhandene Entsprechung in den jeweiligen Werkzeugen erfordern, sich frühzeitig durch geeignete Applikationsprofile auf ein Subset des Standards zu beschränken. AUTOSAR ermöglicht aus unserer Sicht eine evolutionäre Einführung in den heute gelebten Entwicklungsprozess. Das schrittweise Vorgehen bei der AUTOSAR-Einführung erlaubt es, vorhandene Funktionsmodelle, bewährte Prozesse und Werkzeugketten weiter zu verwenden und schrittweise zu optimieren.

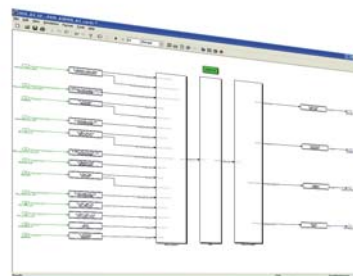


Bild 7: Modelliertes Runnable mit entsprechenden Schnittstellenblöcken nach erfolgter AUTOSAR-Migration.

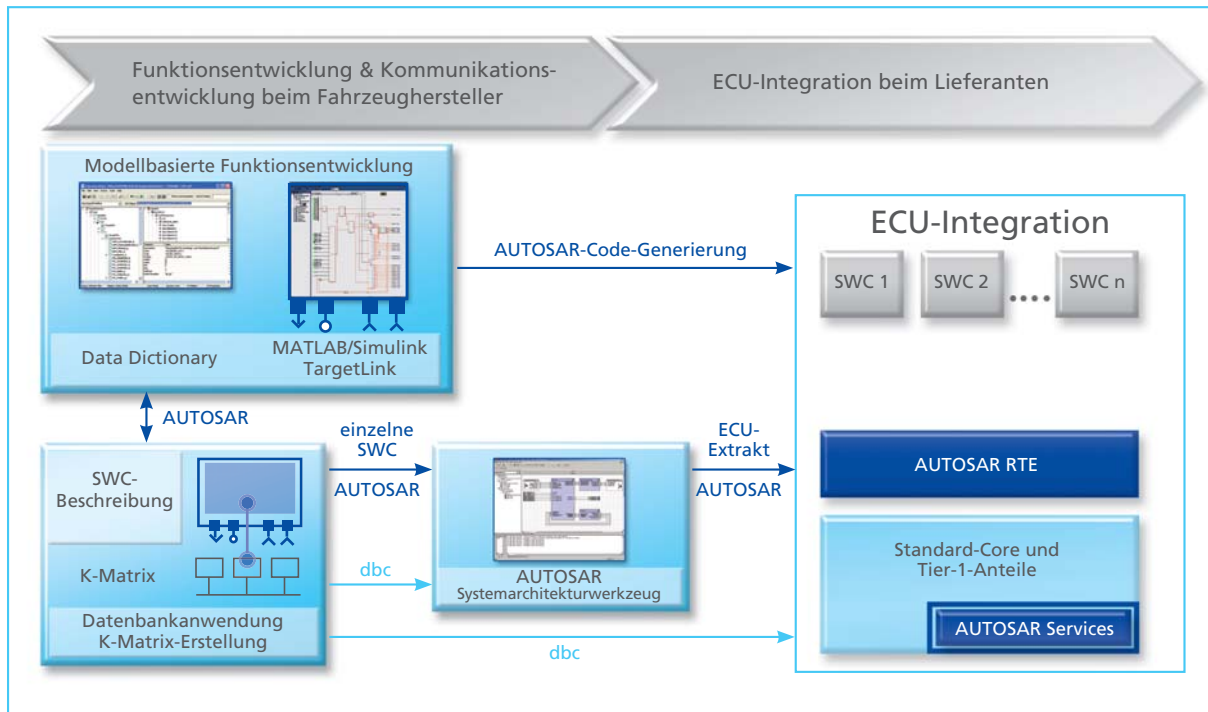


Bild 8: AUTOSAR-Werkzeugumgebung zur Erstellung von AUTOSAR-Softwarekomponenten (SWCs). Der Datenaustausch zwischen den Tools erfolgt auf Basis von AUTOSAR-XML- und DBC-Dateien.

Der erfolgreiche Einsatz der Funktionsmodellierung bei der Entwicklung von Komfort- und Innenraumfunktionen hat sich bei der Daimler AG bewährt. Viele daraus gesammelten Erfahrungen lassen sich auf die AUTOSAR-konforme Entwicklung übertragen und ermöglichen es, dass in Zukunft manuelle Schritte durch automatisierte und standardisierte Prozessschritte ersetzt werden können.

Aufgrund der teilweise noch nicht vollständigen Definition des AUTOSAR-Standards ist eine enge Kooperation mit den Standardisierungsgremien und den Toolherstellern weiterhin notwendig, um so die Umstellung auf den AUTOSAR-Standard zukunftssicher gestalten zu können. ■

*Christian Dziobek  
Dr. Florian Wohlgemuth  
Mercedes-Benz Cars Entwicklung  
Dr. Thomas Ringler  
Group Research & Advanced Engineering  
Daimler AG, Deutschland*

## Fazit

- Die Daimler AG führt AUTOSAR in einem Serienprojekt für Innenraumsteuergeräte ein.
- Die bestehenden Werkzeugketten und Prozesse wurden für AUTOSAR erweitert.
- Bestehende Modelle konnten durch den AUTOSAR-konformen Code-Generator wiederverwendet werden.