

# Systemintegration – im Wechselspiel von Architektur, Technologie und Prozess

10. Jahrestagung Euroforum  
07.02.2006

Dr.-Ing. Günter Reichart  
BMW AG,  
Zentrale Fahrzeugsysteme und –dienste,  
Harald Heinecke  
BMW CAR-IT GmbH

## 1. Einleitung:

Das Elektrik-/ Elektronik-Gesamtssystem eines modernen Kraftfahrzeuges kann zunehmend als ein komplexer Systemverbund von Einzelsystemen betrachtet werden. Sie erbringen in einem Verbund vernetzter Komponenten und Teilsysteme eine breite Vielfalt von Funktionen. Einige Tendenzen dieser Entwicklung zeigt Bild 1.

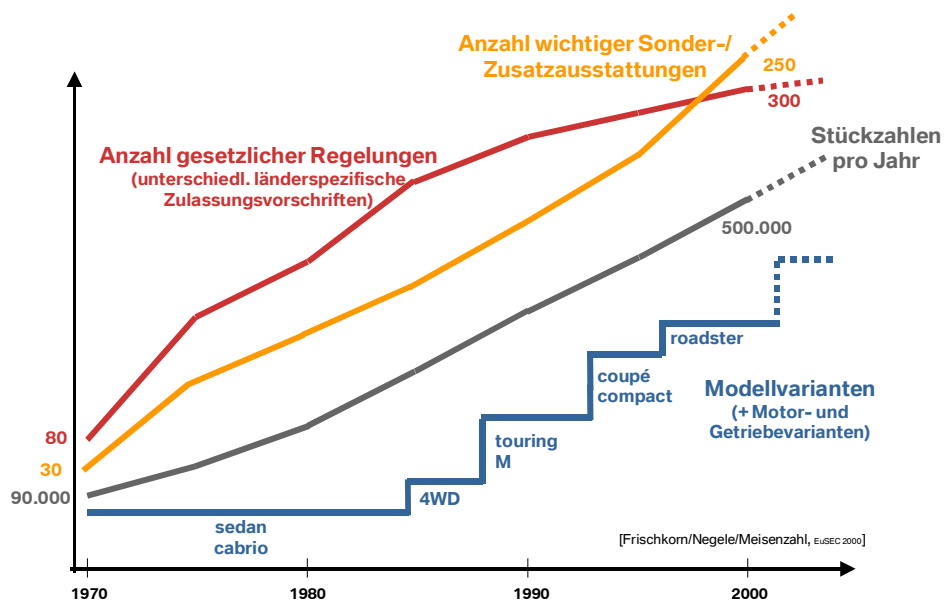


Bild 1: Zunehmende Komplexität der Automobile am Beispiel BMW 3-er Reihe

Es ist unbestritten, dass die Architektur dieses komplexen Gesamtsystems, dessen Entwicklungs- und Verifikationsprozesse wie auch die zum Einsatz kommenden Technologien eine erhebliche Rolle für die Beherrschbarkeit der Systemintegration spielen. Es bestehen durchaus unterschiedliche Auffassungen über die Bedeutung dieser Aspekte und deren Zusammenspiel. So wird beispielsweise die lange Vorlaufzeit der Technologieentwicklung und die Notwendigkeit einer rechtzeitigen Vorentwicklung von Fahrzeuginfrastruktur vielfach noch zu wenig gesehen. Prozesse werden teilweise losgelöst von den technischen Gegebenheiten und Sachzwängen theoretisch formuliert und definiert mit entsprechenden Einführungs- und Akzeptanzproblemen. Eine Konzentration auf „Test to work“ wird in dem zunehmend komplexeren Umfeld immer aufwendiger. Das Ziel des folgenden Beitrages ist es aus Sicht der Autoren den Zusammenhang der genannten Aspekte zu diskutieren und einen Ansatz aufzuzeigen, bei dem Infrastrukturentwicklung, Architekturkonzeption und die Entwicklungs- und Verifikationsprozesse in einen Gesamtkontext gestellt werden und ihre wechselseitigen Bezüge auch im zeitlichen Ablauf aufgezeigt werden.

## **2. Systemintegration als Schlüsselaufgabe der Automobilherstellers**

Es ist inzwischen anerkannt, dass die zentrale Rolle des OEM neben der Produktdefinition in der Systemintegration besteht. Der Begriff Systemintegration wird immer noch vielfach in einem engen Sinne verstanden, der die Integration als das Zusammenfügen der Komponenten und Teilsystemen von Zulieferern zu einem funktionierenden Gesamtsysteme meint. Aus Sicht der Autoren trägt dieses Verständnis von Systemintegration nicht weit genug. Systemintegration beginnt aus unserer Sicht mit dem Systemdesign und damit bereits mit der Technologieauswahl und der Architekturkonzeption.

Unter Systemdesign wird hier der Prozess der Systemauslegung nach funktionalen und nichtfunktionalen Anforderungen verstanden. Hierzu werden die wesentlichen Anforderungen an Hardware und Software in Bezug auf Kundenfunktion, Systemfunktion, Einsatzbedingungen, Energieversorgung und Kommunikation der Systemelemente sowie Kommunalität und Kompatibilität festgelegt. Weiterhin werden Festlegungen der künftigen Steuergerätetechnologien und des physikalischen Bordnetzes getroffen. Im Bereich der Software werden wesentliche Elemente der künftigen

Softwarestruktur, die Wiederverwendbarkeit von SW- Modulen, die Grundstruktur der Software bezüglich HW-Abstraktion und Basisdiensten, wie Netzwerkmanagement, Diagnose oder Flash bestimmt. In dieser Phase werden entscheidende Weichen für die spätere Integrations- und Verifikationsfähigkeit, zum Beispiel hinsichtlich modularer Testbarkeit, gestellt. Fehler die in dieser Phase gemacht werden, lassen sich, wenn überhaupt, in späteren Phasen meist nur mit hohem Aufwand korrigieren. Aus diesen frühen Festlegungen ergeben sich die Architekturelemente anhand derer der Systemarchitekt unter Betrachtung der vielfältigen Anforderungen an Kunden- und Infrastrukturfunktionen Lösungskonzeptionen entwerfen und bewerten kann.

Systemdesign und Entwicklung der Systemarchitektur, haben sich traditionell eher aufgrund von bottom up Vorgehensweisen und über viele Fahrzeugprojekte gesammelte Erfahrungen entwickelt und werden meist noch als eine für ein bestimmtes Fahrzeugprojekt durchzuführende Entwicklungsaufgabe verstanden. Es zeigt sich aber zunehmend, dass sich eine Technologiefestlegung und Architekturkonzeption auf einen längeren Betrachtungshorizont erstrecken muss, der nicht nur ein einzelnes Produkt oder eine einzelne Produktlinie im Fokus hat. Notwendig ist eine längerfristig angelegte, produktlinienübergreifende Technologie- und Architekturstrategie und eine damit einhergehende Prozessentwicklung.

## **2.1 Die Rolle der Technologie**

Technologien besitzen für die E/E-Systemarchitektur moderner Fahrzeuge eine Doppelrolle. Auf der einen Seite sind sie Befähiger, die neue Möglichkeiten und Freiräume schaffen, etwa durch höhere Bandbreiten, neue Protokollmerkmale, neue Übertragungsprinzipien, oder neue Lösungskonzepte eröffnen, wie beispielsweise Supercaps. Auf der anderen Seite sind Technologien Veränderungstreiber, wenn beispielsweise veraltete Technologien auslaufen und Entwicklungsprozesse angepasst werden müssen.

Die Technologieentwicklung, das heißt die Entwicklung neuer Kommunikationsprotokolle, neuer physikalischer Busse waren in der Vergangenheit vielfach von der Initiative einzelner Firmen, sei es von der Zuliefererindustrie, sei es von den OEM's abhängig. Eine systematische, firmenübergreifende langfristige Technologieentwicklung fand nur partiell statt.

Wiederholt kam es deshalb zu Parallelentwicklungen mit teilweise langwierigen Harmonisierungsprozessen (CAN vs. VAN, TTP vs. FlexRay) oder zu nur kurzfristig lebensfähigen proprietären Lösungen (Byteflight, D2B).

Notwendig wird aber in Zukunft eine firmenübergreifende, rechtzeitige Technologieentwicklung und de-facto Standardisierung. Die aktuellen Beispiele FlexRay und AUTOSAR zeigen den hohen Wert einer firmenübergreifenden de-facto Standardisierung. Ohne diese frühe und intensive Zusammenarbeit wäre auch für einen einzelnen Automobilhersteller der Umstieg auf derartige neue technologische Ansätze nicht mehr leistbar. Natürlich müssen einzelne Firmen hier eine Führungs- und Treiberrolle übernehmen, dazu gehört auch die Bereitschaft andere Partner an den erzielten Ergebnissen, soweit nicht vertraulich, teilhaben zu lassen.

## **2.2 Die Definition und Findung der Architektur**

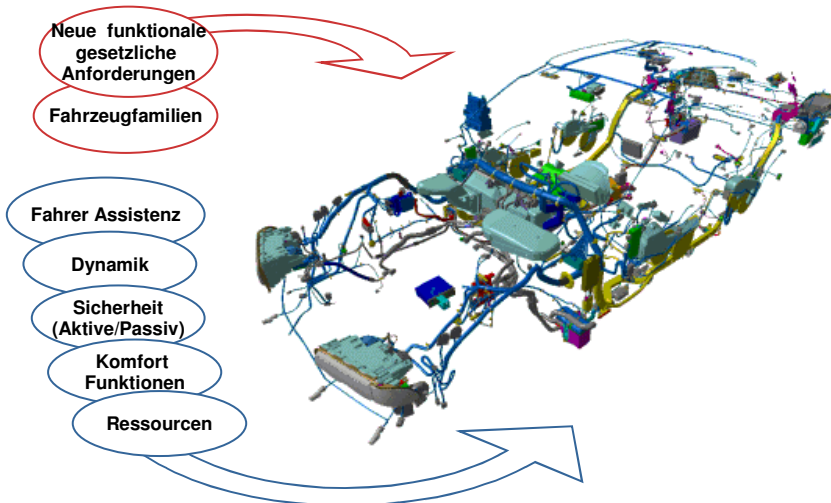
Zwar ist eine exakte Definition dessen, was unter einer E/E-Systemarchitektur im Kraftfahrzeug zu verstehen sei, noch nicht allgemein etabliert. Für diesen Beitrag verstehen wir unter einer Systemarchitektur die Struktur eines Systems hinsichtlich

- des Zusammenwirkens und der Vernetzung der Systemelemente
- der funktionalen Beziehungen der Systemelemente
- der Schnittstellen zu anderen Systemen
- der Umgebung
- des Datenflusses im System
- der Daten- und Softwarearchitektur

Unstrittig ist, dass es auf eine Systemarchitektur unterschiedliche Sichten gibt. So wird primär die logische Sicht, als Sicht der funktionalen und logischen Beziehungen der Funktionselemente, die technische Sicht als Sicht auf die konkrete technische Realisierung u.a. in Form von Bussen und Steuergeräten und die topologisch/geometrische Sicht, als Sicht auf die strukturelle und räumliche Anordnung der Steuergeräte unterschieden. Je nach Art der Sicht auf sie Systemarchitektur rücken eher prozessuale und methodische Aspekte, wie modellbasierte Entwicklung, technologische Aspekte, wie Bustechnologien,

strukturelle Merkmale, wie Ringstruktur, Busstruktur oder Sternstruktur, oder packageorientierte Aspekte in den Vordergrund.

Ein ganz zentraler Aufpunkt für die Auslegung der Architektur besteht in Festschreibung der technischen Einflussfaktoren auf die Systemauslegung. Hierzu müssen neben abzubildenden Kundenfunktion auch die nichtfunktionalen Anforderungen sowie die Gültigkeitsbereiche über Fahrzeugfamilien und Lebenszyklus definiert sein. ( Bild 2).



*Bild 2: Anforderungen - Eingangsgrößen*

Diese nahezu zeitlich explodierenden Anforderungen sind sauber zu analysieren und in einer allerersten Stufe muss eine Grundtechnologie Auswahl stattfinden: Was wird mechanisch, was elektrisch, was elektronisch und was über elektronische Vernetzung realisiert? Dabei ist zu berücksichtigen, dass eine später eingeplante elektronische Vernetzung für eine höherwertige Funktion durchaus dann schon in der Grundstufe eine elektronische Vernetzung unterstützen muss.

Für die Verteilung der elektronisch vernetzen Funktionen im Fahrzeugnetzwerk ist eine stabile Infrastruktur zwingend. Nur auf dieser festen Basis einer Software Infrastruktur kann ein offener Partitionierungsablauf aufsetzen. (Bild 3) .

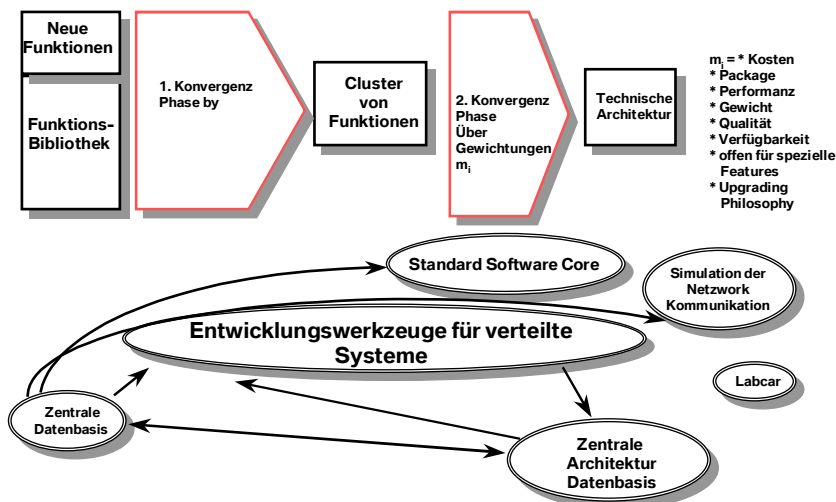


Bild 3: System Design Prozess

Für die Architekturfindung ist dann ein Konvergenzprozess möglich, der im Sinne einer mehrdimensionalen Kostenfunktion gegenüber den oftmals konkurrierenden Wichtungsfaktoren wie, Singularitäten der Kommunikation, Implementierungskosten, Performanz, Verkabelung, Gewicht, Bauraum, Erweiterbarkeit,...ein Optimum zu finden hat. Wesentlich ist dabei, dass die Infrastruktur und die ausgewählten Technologien in nahezu serienreifen Zuständen bei dem Umsetzungsstart nach erfolgter Architekturfindung vorliegen. Denn: eine spätere Änderung von grundlegenden Technologien wird eine absolut neue Optimum Lösung bezüglich der Architektur bewirken oder gar den Lösungsraum vernichten. Ein gutes Beispiel ist dabei die verfügbare Bandbreite eines Kommunikationssystems. Ein Überschreiten der maximal zulässigen Bandbreite wird hier sicherlich eine deutliche Design Änderung hervorrufen. Busse mit höherer Übertragungsbandbreite und leistungsfähigere Steuergeräte erlauben eine höhere Integration von Funktionen und eine andere Partitionierung von Funktionen als dies bei weniger leistungsfähigen Systemen möglich wäre. Analoges gilt für die Art der Softwarearchitektur. Standardisierte Schnittstellen, eine Middlewareschicht und eine Hardwareabstraktionsschicht, wie in AUTOSAR verfolgt, erleichtern die Verschiebbarkeit von Funktionen zwischen Steuergeräten.

Die enge Wechselbeziehung zu den zum Einsatz kommenden Technologien gilt sowohl für das Datenbordnetz wie auch für das Energiebordnetz. Standen am Beginn des Elektronikeinsatzes im Kraftfahrzeug zunächst einzelne lokal abgegrenzte Anwendungen, so hat sich mit dem zunehmenden Einsatz von Elektronik, der Vielzahl von Steuergeräten und der Einführung von Bussystemen das Bild komplett gewandelt. Mehr als 60 elektronische Steuergeräte unterschiedlicher Zulieferer übernehmen die Steuer- und Regelungsaufgaben eines modernen Oberklassefahrzeuges (Bild 4). Sie kommunizieren untereinander und mit den Sensoren und Aktuatoren über verschiedene Bussysteme, wie CAN, MOST, Byteflight, LIN oder in naher Zukunft auch Flex Ray

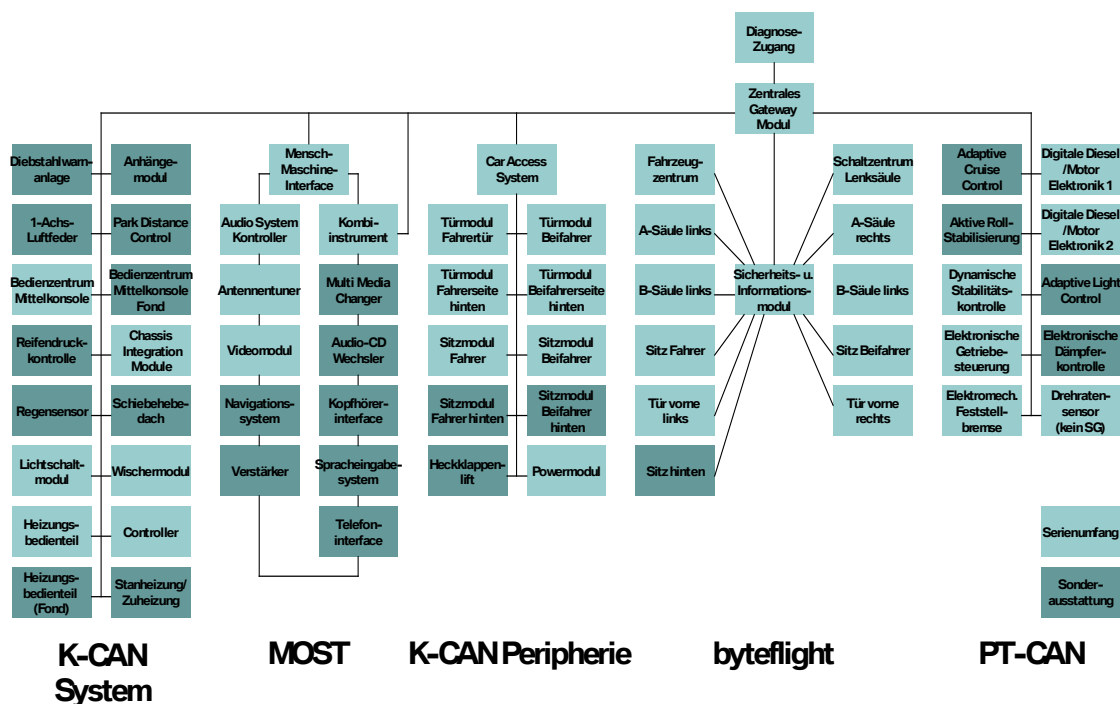


Bild 4: Steuergeräteverbund in einem modernen Fahrzeug der Oberklasse

Die Funktionalität eines modernen Kraftfahrzeuges lässt sich nur mit Hilfe eines umfassenden Einsatzes von Elektronik realisieren, da gerade die Innovationen nur über funktionale Vernetzung darzustellen sind. Es ist evident, dass sich damit

nicht nur erhebliche technologische Veränderungsimpulse ergaben, sondern auch vermehrt Veränderungen der Entwicklungsprozesse notwendig waren und weiter sind.

Die Abstimmung der Aufgaben zwischen den verschiedenen Partnern aus der Zuliefererindustrie und die Sicherstellung gleichartiger und gleichwertiger Entwicklungs- und Verifikationsprozesse bei Hardware – und Software ist eine zentrale Rolle des OEM. Es hat sich als sehr zielführend erwiesen, dass der OEM beispielsweise die Art und Variabilität der in den Steuergeräten eingesetzten Prozessoren, ASIC's, FPGA's und Transceiver erheblich mitgestaltet und auch entsprechende Qualifizierungsmaßnahmen für die eingesetzte Hardware vorgibt. Anders wäre die Sicherstellung der hohen Steuergerätequalität und die Bereitstellung von Standard Core Lösungen für die Basis-Software schon längst nicht mehr beherrschbar.

Die Zahl der Steuergeräte hat allein aus Packagegründen eine Grenze erreicht. Auch aus Zuverlässigkeitsgründen ist eine stärkere Integration der Funktionen in weniger, aber leistungsfähigeren Steuergeräten wünschenswert. Dem entgegen stehen jedoch die breite Spreizung des Ausstattungsumfanges und der Wunsch nach einem individualisierbaren Funktionsangebot. Weiterhin muss die Funktionsverfügbarkeit bei stärkerer Zentralisierung sichergestellt bleiben, was zu stark steigenden Anforderungen an die Zuverlässigkeit der Steuergeräte und der angesteuerten Aktuatoren führt. Auch unterschiedliche Anforderungen an die Funktionssicherheit und Verfügbarkeit von Funktionen in einem Steuergerät setzen der Hochintegration hier Grenzen.

Die Stromversorgung der Elektronik, insbesondere aber der Leistungsverbraucher erfordert heute ein extrem leistungsfähiges Energiebordnetz und ein ausgefeiltes Energiemanagement, um neue Funktionen für geringeren Verbrauch und reduzierte Emissionen, wie Motor-Start-Stop-Systeme oder elektrifizierter Antriebsstrang zu ermöglichen. Zunehmend entsteht damit auch für das Energiebordnetz ein erheblicher Komplexitätsanstieg. Mehrspannungsbordnetze mit Hochvoltanwendungen bis zu 300V, unterschiedliche Speichertechnologien, DC/DC-Wandler erweitern das Spektrum der zum Einsatz kommenden Komponenten und stellen auch neue Anforderungen an Handlings-, fertigungs-



und Serviceprozesse. Komplexe Wechselwirkungen zwischen Spannungslagen und Betriebszuständen des Bordnetzes mit diversen Teilssystemen (z.B. Klimakompressor, Multimediasystemen) müssen beachtet und durch geeignete Maßnahmen in der Auswirkung begrenzt werden.

Der Prozess der Architekturkonzeption erfolgt bei BMW in mehreren Stufen (s. Bilder 2 und 4). Anfangs werden Grundprämissen der Architektur, wie der Einsatz bestimmter Technologien oder Fragen der Rückwärtskompatibilität, diskutiert und zu einer Grundsatzentscheidung gebracht, anschließend erfolgt ein iterativer Abstimmprozess zwischen den Domänen Chassis, Antrieb und I&K mit der zentralen Systemarchitekturstelle in einem Architekturteam, in dem Anforderungen abgestimmt und Lösungskonzepte auf ihre Tragfähigkeit und Konsensfähigkeit geprüft werden. Der Fortschritt der Architekturdiskussion wird in sogenannten Design Decision Documents festgehalten, die den späteren Nachvollzug getroffener Konzeptentscheidungen ermöglichen. Die Architekturstelle gibt zu definierten Zeitpunkten sogenannte Architekturstände vor, die den aktuell verbindlichen Arbeitsstand repräsentieren. Den Abschluss findet die Architekturentwicklung im sogenannten Architekturfreeze. Änderungen, die nach diesem Termin erfolgen müssen, bedürfen spezieller Änderungsanträge und werden in einem eigenen Prüfverfahren auf Notwendigkeit und Auswirkungen geprüft.

## Architekturdesignprozess Kernprozesse

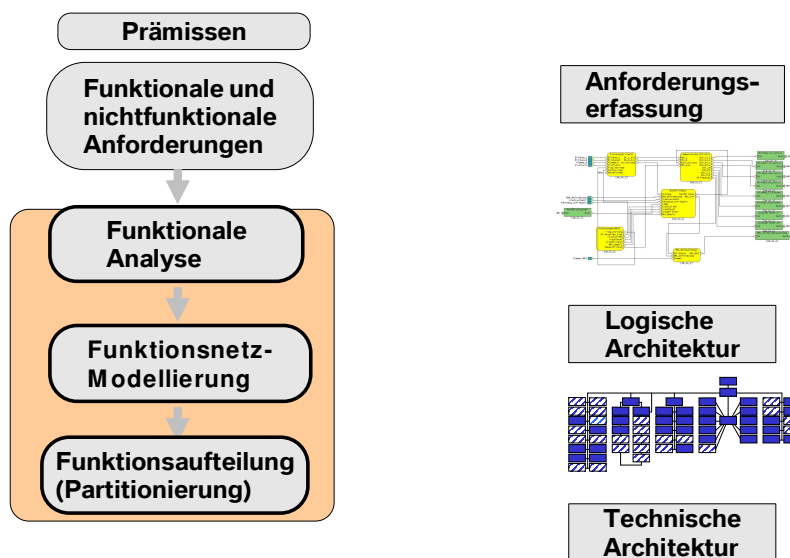


Bild 4 Kernprozess im Architekturdesignprozess

Die Fragen des Architekturlebenszyklus wurden bisher oft zu wenig beachtet, obwohl fast jede Systemarchitektur nicht nur über viele Jahre in einer Fahrzeugbaureihe besteht sondern in gleicher oder modifizierter Struktur in weiteren Baureihen zum Einsatz kommt. Die Architektur muss daher von ihrer Skalierbarkeit an die in Frage kommenden Baureihen anpassbar sein, andererseits für die in dem Lebenszyklus zu erwartenden Innovationen entsprechend erweiterbar und ggü. zu erwartenden technologischen Veränderungen auch möglichst robust ausgelegt sein. Eine Architektur, die zum Beispiel Bustechnologien verwendet, die schon heute an die Grenze ihrer Leistungsfähigkeit kommen, kann diesen Anspruch an Technologierobustheit nicht erfüllen. Dieser Anspruch nach Erweiterbarkeit und Zukunftsfähigkeit steht im Zielkonflikt mit den Anforderungen nach Wiederverwendbarkeit von Komponenten, nach Rückwärtskompatibilität und nach Baukastenlösungen. Der Zielkonflikt kann nur im Sinne eines besten Kompromisses bewältigt werden. Entscheidend hierfür ist, dass die Tradeoffs die sich für verschiedene Lösungsoptionen ergeben klar und nachvollziehbar herausgearbeitet werden und nicht nur eine kurzfristige Optimierung auf eine Produktlinie erfolgt. Jeder Entwurf einer Systemarchitektur muss einerseits auf die Technologien zurückgreifen, die zu diesem Zeitpunkt verfügbar sind, andererseits kann und wird ein Architekturentwurfsprozess Aspekte aufwerfen, die es erfordern, nach technologischen Lösungsmöglichkeiten zu suchen. Welches Ausmaß an technologisch neuartigen Lösungen in einer Architektur Berücksichtigung finden, ist eine äußerst schwierige Entscheidung. Hier hilft nur ein offener Diskussionsprozess und ein transparentes Risikomanagement, denn eine konservative Strategie verfehlt das Ziel der Technologierobustheit, eine zu progressive verfehlt das Qualitäts- und Kostenziel und vielfach auch das Terminziel. Ein weiterer wesentlicher Aspekt in der Bewertung und Optimierung von Architekturkonzepten liegt in der Frage der Auswirkung auf die Implementierung, d. h. ihre Umsetzung in der Entwicklung, Produktion oder im Service. Stehen wesentliche Entwicklungsmethoden und -werkzeuge nicht rechtzeitig zur Verfügung, sind „handgestrickte“ Lösungen oder umfangreiche Schulungen der Entwickler in der heißen Entwicklungsphase notwendig. Sind neue Diagnoseverfahren in Produkten oder Service erforderlich, ergeben sich

auch hier hohe Risiken für die Systemintegration. Auch hier muss ein konsequentes Risikomanagement aufgesetzt werden, das die Initiierung und Realisierung der notwendigen Arbeiten betreibt und überwacht. Gegebenenfalls sind Alternativszenarien zu beplanen und Entscheidungszeitpunkte für Alternativlösungen festzulegen..

## Zukünftige E/E-Systemarchitekturen

### Lösungselemente

- Hierarchisierung
  - Zentrale Systemfunktionen
  - Konsequentes Energiemngmt.
  - Central sensor data fusion
  - Koord.-instanzen ICM, IPM
- HW-Abstraktion
  - HIS, AUTOSAR
- Modularität u. Verschiebbarkeit
  - AUTOSAR
  - SW Funktionsbibliotheken
  - Modulare Testbarkeit
- Transparenz des dynamischen Verhaltens
  - FlexRay
  - Spezifikation des dynamischen Schnittstellenverhaltens
- Reduktion der Heterogenität
  - Einheitliche Gateways
  - Reduzierte Subbus Varianten
  - Klare Migrationsstrategien
  - FlexRay
- Standardisierung der Basis-SW
  - AUTOSAR
- Standardisierung der Schnittstellen
  - AUTOSAR
- Entwicklungsprozess
  - Modellgestützte Entwicklung
  - Funktionsnetzwerke
  - Systemorientierung
  - Proaktive Planung (Obsolescence, Technologie-updates,...)
  - Architekturbewertung

*Bild 5 Lösungselemente von E/E Systemarchitekturen*

### 2.3 Die Rolle der Prozesse

„Prozesse folgen Technologien und Produkten und nicht umgekehrt.“ Diese alte Regel drückt eine wichtige Erkenntnis aus. Prozesse dürfen niemals Selbstzweck werden, sondern müssen dem Ziel dienen, den Entwicklungsablauf im Hinblick auf die Entwicklungs- und Realisierung des Produktes zu optimieren. Dies kann nur erreicht werden, wenn Projekterfordernisse, technische Gegebenheiten und Sachzwänge wie auch Ressourcen in einer Gesamtbetrachtung bestmöglich im Zusammenspiel gestaltet werden.

Prozesse benötigen eine ständige Anpassung und Weiterentwicklung, da sie stark von den eingesetzten Technologien, von verfügbaren Tools aber auch von jeweils herrschenden Entwicklungsparadigmen geprägt werden. Es nutzt beispielsweise wenig einen perfekten Testprozess definiert zu haben, wenn die notwendigen Test- und Analysetools nicht rechtzeitig in ausreichender Qualität zur Verfügung stehen. Insoweit ist also bereits in der Technologienentwicklung

begleitend die Auswirkung auf die spätere Systemintegration zu analysieren. Entsprechende Methoden und Tools wie auch Prozessabläufe sind bereits weitgehend parallel zur Technologienentwicklung aufzusetzen. Dies hat sich gerade in der Entwicklung der Flex Ray Technologie erneut bestätigt. BMW hatte intern eine Reihe von Projektaktivitäten im Sinne der Prozess-, Methoden- und Toolentwicklung parallel zu der Technologieentwicklung gestartet ohne die die Umsetzung von Flex Ray in den Serienapplikationen nicht möglich gewesen wäre. Dennoch bleiben hier für die Zukunft noch erhebliche Aufgaben. Wie beispielsweise ein Systemdesignprozess für deterministische Systeme genau gestaltet werden muss, ist noch Gegenstand der Entwicklung. Ähnliches gilt auch für das Projekt AUTOSAR. Eine veränderte Softwarearchitektur der Steuergeräte erfordert auch veränderte und angepasste Prozesse. Dies wird in entscheidendem Maße auch davon abhängen, welche Bereitschaft zur Offenlegung funktionaler Schnittstellen in den Applikationsmodulen erreichbar sein wird. Hiervon wird beeinflusst, in welcher Granularität Applikationsmodule als separierbare Umfänge konzipiert werden können, welcher Grad an Verschiebbarkeit von Softwaremodulen erreichbar sein wird und welche Form der Hinterlegung solcher SW-Module in einer Softwarebibliothek möglich sein werden.

Wie eine AUTOSAR- basierte SW-Entwicklung im Rahmen einer breiten Anwendung zwischen Zulieferern und OEM's praktisch gelebt werden kann wird

sich erst mit konkreten Pilotapplikationen näher bestimmen lassen.

## Standardization

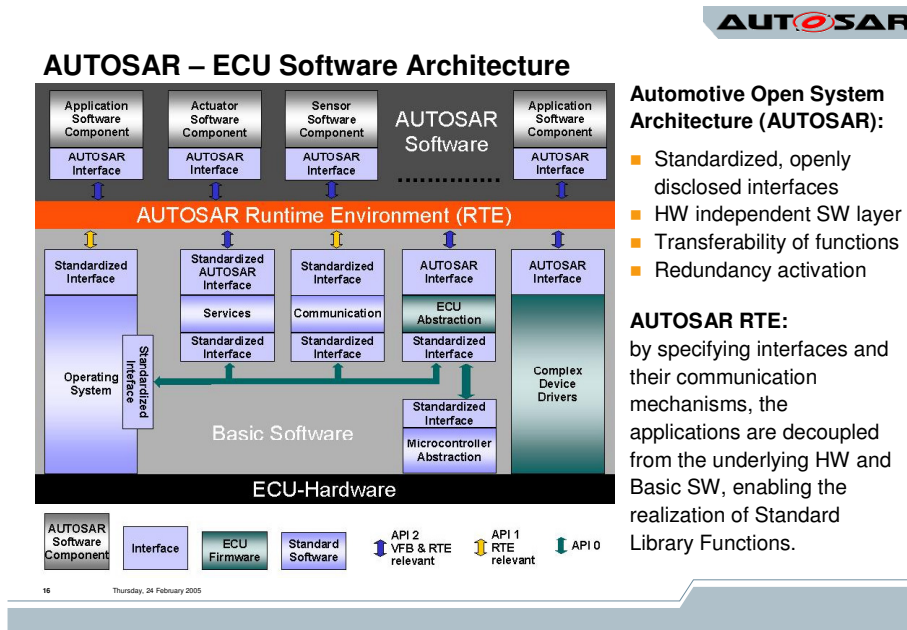


Bild 6: Die AUTOSAR Architektur ( entnommen aus Ref. [1,2])

Bild 6 zeigt, wie bei AUTOSAR die grundlegende Software Infrastruktur Architektur definiert wurde. Ein wichtiges Element ist dabei das Run time environment (RTE), das ECU intern die Kommunikation regelt. Es ist sofort einsichtig, dass eine Integration einer derartigen RTE Konsequenzen auf den Systemdesignprozess haben muss, da eine Konfiguration der RTE erforderlich ist. Dieser AUTOSAR Designprozess wird durch eine Reihe von Spezifikationsschritten aus einem AUTOSAR Prozess unterstützt (Bild 7).

# The AUTOSAR Method

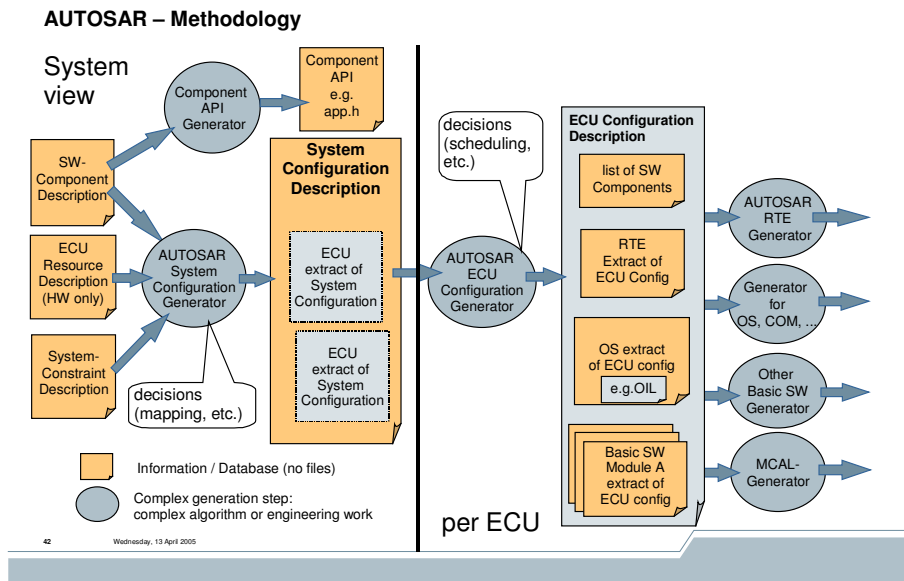


Bild 7: Der AUTOSAR Design Prozess ( entnommen aus Ref [3])

SW- Komponenten - , ECU Ressourcen – und System Beschreibungen – Templates sind wesentliche Eingangs Größen, um eine Systemkonfiguration zu erhalten, d. h. es werde n die SW Komponenten und damit Ensembles von Funktionen auf Steuergeräte abgebildet. Der nächste Schritt fokussiert dann auf das individuelle Steuergerät, wobei die Generation der entsprechenden SW Komponenten direkt angestoßen wird.

Bild 8 zeigt nun wie stark ein Generierungsprozess von z. Bsp. Der RTE von den Anforderungen abhängt: Im originären AUTOSAR Design (Bild 8) erfolgt die Generierung direkt aus der Bestimmung der Partionierung, was gerade bei „Erst Design Aufgaben“ wesentlich ist. Will man allerdings Altkomponenten neu auflegen oder in neue Fahrzeugnetzwerke einbinden, so ist auch ein Designprozess, der von einer existierenden Kommunikationsmatrix ausgeht zu unterstützen.

### **AUTOSAR Design:**

Von der Partitionierung zur Kommunikations Matrix.



### **legacy Partitionierung:**

Vom Nachrichtenkatalog zu ECU Signalen



### **Einführungsanforderung:**

„Tooling und Prozess: Design Pfad von existierender Komm-Matrix zur AUTOSAR Kommunikation“

Bild 8: Der Prozess hat die Anforderung zu erfüllen (*Details s. Ref [4]*)

Ähnliche Anforderungen von festen Beziehungen auszugehen kommen auch aus dem Bedarf dem System deterministische Eigenschaften aufzuprägen. Dies kann getriggert sein durch Anwendungen aus einem Sicherheits - relevanten Umfeld oder durch stringente Anforderungen in Bezug auf Zusammensetzbarkeit. Bild 9 zeigt, dass für die Entwicklung deterministischer Systeme es wesentlich ist synchrone Kommunikationssysteme einzusetzen. Das scheduling ist dann schon in ganz frühen Phasen des Designs in den Templates ( s. Bild 7) vorzugeben. Bild 9 verdeutlicht die Synchronisation der Sensorwerte der beteiligten Steuergeräte aus einem Kommunikationszyklus  $m$  und die synchronisierte Erfassung der Werte dieses Zyklus  $m$  zu Beginn des Zyklus  $m+1$  auf Applikationsebene in den Steuergeräten. Daraus folgt dann die Berechnung der Sensorwerte und Stellgrößen. Dieser Determinismus führt dann zu einer Weitergabe /Ausgabe der Stellgröße während dieses nachfolgenden Kommunikationszyklus  $m+1$ . Ein derartiges deterministisches Design wird immer wesentlicher für die Darstellung hoch vernetzter Funktionen. Es wird in Zukunft die notwendige Voraussetzung für die gewünschte freie Partitionierbarkeit und damit Zusammensetzbarkeit von Funktionalitäten darstellen.

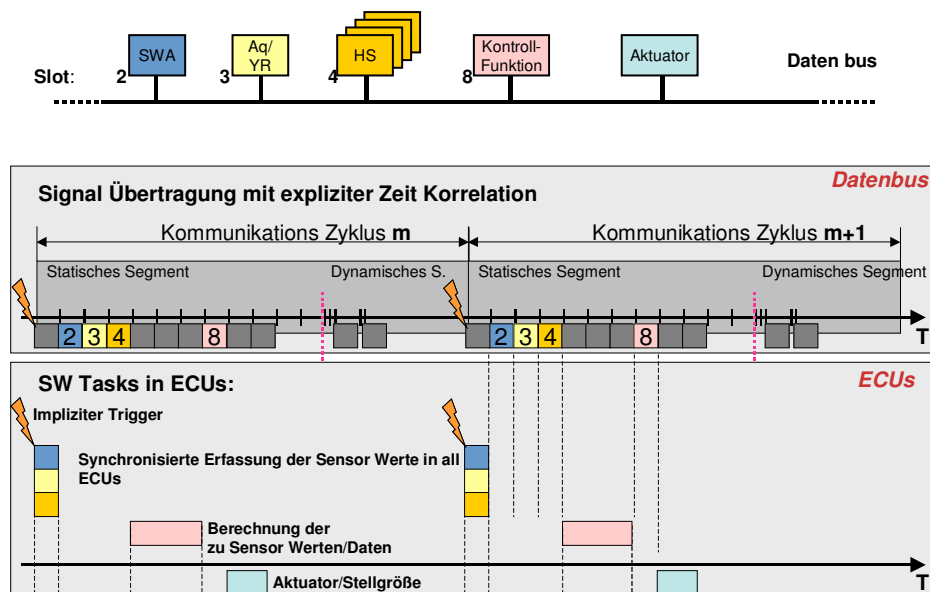


Bild 9: Synchronisation von SW, Sensor und Aktuator Tasks ( s. a. Ref [5])

Die Autosar Technologie schirmt die Applikation und damit die Anwendungssoftware von den Hardware Eigenschaften weitgehend ab. An festen Terminen ( MS in Bild 10) sind die Entscheidungen im Systemdesign einer Fahrzeugserie zwingend welche Infrastrukturtechnologien eingesetzt werden. Durch Einsatz einer RTE kann jedoch die oberste Schnittstelle AUTOSAR konform implementiert werden, auch wenn in den unteren Schichten noch nicht alle AUTOSAR Module serienreif verfügbar sein sollten. Später, bei Verfügbarkeit aller AUTOSAR Implementierungen ist dann ein Migrationsprozess zu vollständig konformen AUTOSAR zugänglich.



Migration scenarios allow concurrent deployment of AUTOSAR modules and existing proprietary components to one ECU.

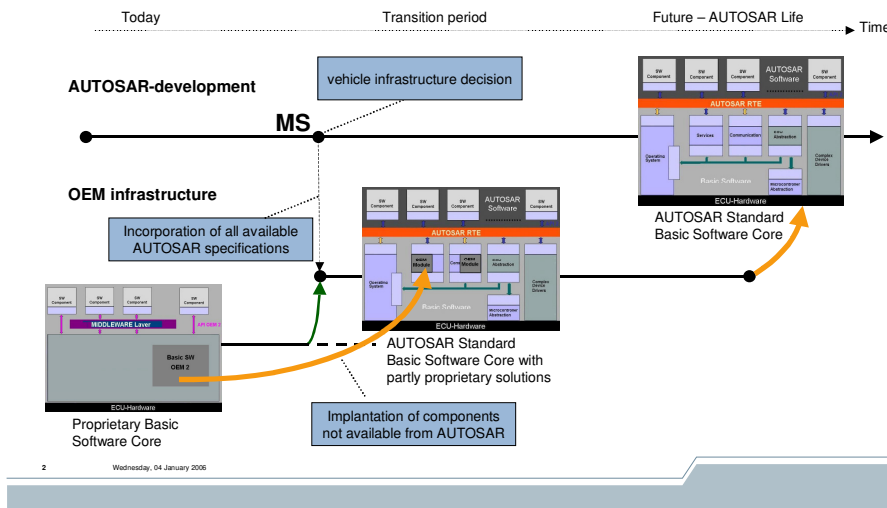


Bild 10: Fahrzeugserien haben vor Serienentwicklungsstart die Infrastruktur festzuschreiben, Optionen zu Migration ( AUTOSAR Vortrag Baden – Baden September 2005)

Bild 11 verdeutlicht den bei BMW gefahrenen Migrations Prozess unter Bereitstellung der Standard Software Core Implementierungen SC 6.2 bis SC 7.0.

In ein existierendes Umfeld von Standard Software [6] wird eine so genannte single sided Autosar runtime environment eingebracht, d. h. die Schnittstelle zur Infrastruktur ist nach „unten“ zum BMW Standard core noch proprietär, während die „obere“ Schnittstelle zur Applikation hin schon AUTOSAR konform ausgelegt wird. Eine weiter Einbindung von AUTOSAR konformen Infrastruktur Modulen hat so zu einer Version 6.2 geführt. Der kontinuierliche Ersatz von proprietären Modulen führt dann zu einer am Ende stehenden 100% AUTOSAR kompatiblen Infrastruktur (V7.0). Sicherlich muss in diesem Prozess die untere Schnittstelle der RTE immer wieder anpasst werden, aber man vermeidet ein starke Anpassungstätigkeit and den Schnittstellen der Applikationsmodule. Dieses Procedere vermeidet einen Big Bang bei der Technologie Umstellung und ermöglicht ein abgesichertes Risikomanagement.

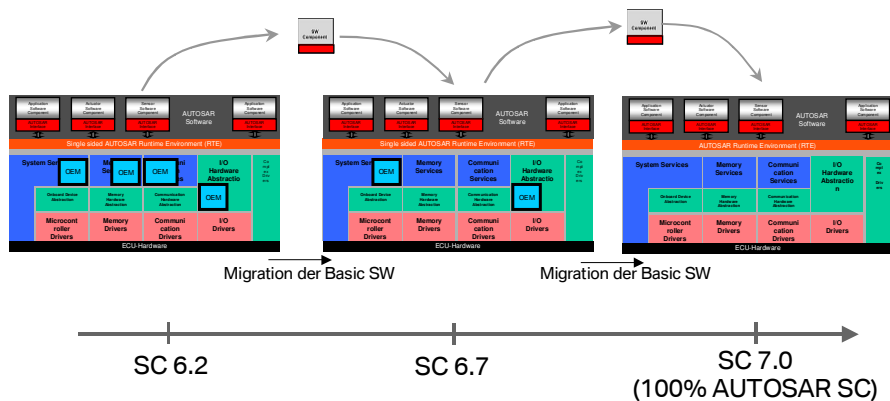


Bild 11: Migrationsunterstützung über Standard Software Releases (Details s. Ref [4])

### 3. Systemintegration in Zukunft

Der Begriff Systemintegration ist, wie bereits eingangs erwähnt, nicht klar definiert und unterliegt einem sehr unterschiedlichen Verständnis. Teilweise wird lediglich die Aggregation von Komponenten zu einem arbeitsfähigen System darunter verstanden:

“Assembling many components so they can work together as a system.”

Systemintegration ist aber mehr als das Zusammenführen und Verifizieren von Komponenten oder Teilsystemen. An anderer Stelle wird schon etwas präziser der Prozess des Zusammenführens und Testens von Systemkomponenten zu einem umfassenden, interoperablen Gesamtsystem darunter verstanden:

“The progressive linking and testing of system components to merge their functional and technical characteristics into a comprehensive, interoperable system. Note: Integration of data systems allows data existing on disparate systems to be shared or accessed across functional or system boundaries”

Schließlich findet sich auch eine Definition, die unserem Verständnis von Systemintegration nahe kommt. Systemintegration wird hier als der Prozess der

Entwicklung und Verifikation komplexer IT-Systeme vom Systemdesign über die Architekturkonzeption bis hin zur Gesamtsystemintegration verstanden:

“The creation of complex information systems that may include designing or building a customized architecture or application, integrating it with new or existing hardware, packaged and custom software, and communications”

In unserem Verständnis sollte E/E-Systemintegration für Kraftfahrzeuge als Gesamtprozess zur Konzeption, Entwicklung und Verifikation von komplexen Elektrik-/Elektroniksystemen verstanden werden, welche über Anwendungsbereiche/Fahrzeuge hinweg wieder verwendbar sind.

Systemintegration erfordert eine Gesamtbetrachtung von Anforderungen, Technologien, Architekturen und Prozessen. Dies bedeutet nicht, dass dies auch organisatorisch in einer Einheit abgebildet werden müsste. Vielmehr müssen alle hierfür notwendigen Kompetenzen zusammengeführt und in engen Dialog gebracht werden.

Insbesondere die Technologievorentwicklung im Bereich der Bordnetzinfrastruktur, die Entwicklung neuer Software-Architekturen, die Entwicklung zentraler Systemfunktionen, wie Diagnose, Fahrzeug-Security oder Fahrzeugprogrammierung, und zentraler Steuergeräte erfordert eine enge Zusammenarbeit zwischen Systemarchitekt, SW-Architekten, Infrastrukturentwicklern und Domänenarchitekten. Nur aus dem Zusammenführen unterschiedlicher Sichten lassen sich hier langfristig tragfähige Lösungen entwickeln. Kernstück ist eine gemeinsame produktlinienübergreifende Technologie- und Architekturstrategie, die die Leitlinie der künftigen Entwicklungen beinhaltet und für eine Koordination von Teilaktivitäten sorgt. Andernfalls besteht die Gefahr, dass domänenlokale Gesichtspunkte dominieren, Strategien auseinander laufen und Prozesse zu sehr auf lokale Anwendungsinteressen fokussiert bleiben.

Heutige Konzepte für Systemintegration betonen unter dem Schlagwort „Funktionsorientierung“ eine funktionale, meist hierarchische Dekomposition der Gesamtfunktionalität, d.h. die Zergliederung in Hauptfunktionen, Unterfunktionen und Detailfunktionen. Die Dekomposition der Funktionen und die Beschreibung der funktionalen Zusammenhänge in einer Modellierung des Funktionsnetzwerk hilft die wesentliche Abhängigkeiten zwischen Funktionen zu erkennen. Dieser Weg einer funktionsorientierten Entwicklung ist dem traditionellen eher

komponentenorientierten Entwicklungsparadigma überlegen. Er fördert die Vorgehensweise, zunächst die Funktionen und ihre Eigenschaften zu definieren, die resultierende Funktionsstruktur zu bestimmen, Neben- und Randbedingungen zu definieren und damit den technischen Lösungsraum ohne zu frühe Vorfestlegungen einzugrenzen. Dieser Prozess wird durch die modellbasierte Entwicklung, Funktionsnetzwerkanalyse auf Basis von Matlab oder UML 2.0, sehr wirksam unterstützt. Eine der Herausforderungen in Zukunft besteht, darin dass die Modelle die in unterschiedlichen Phasen des Entwicklungsablaufes entstehen, z. B. im Rahmen der Funktionsnetzmodellierung, im Rahmen der Komponentenetwicklung oder im Rahmen einer modellbasierten Verifikation, nicht auseinander laufen, gleiche Sachverhalte nicht mehrfach erneut modellieren oder anders modellieren. Modelle müssen dabei entweder von den Entwicklern des OEM oder von den denen der Zulieferer nach einheitlichen Modellierungsregeln erstellt und zugeliefert werden. Die Akzeptanz dieser Vorgehensweise kann aber nur erreicht werden, wenn auch ein Nutzen durch erleichterte Integration in späteren Phasen beim Entwickler ankommt.

Insbesondere der derzeit diskutierte Weg zu Software-Funktionsbibliotheken für wieder verwendbare Softwaremodule erfordert klare Festlegungen, welche Funktionen in welcher Modularität wie beschrieben in einer solchen Funktionsbibliothek hinterlegt werden. Eine Weiterentwicklung dieses Ansatzes zielt dann Richtung sogenannter Software-Productlines, die die effiziente und systematische Erzeugung von Softwarelösungen auf Basis definierter Core Assets für definierte Anwendungsbereiche erlauben. Auf Architekturebene wäre es längerfristig ebenfalls sinnvoll, bestimmte und bewährte Lösungsmuster für typische Fragestellungen als sogenannte architectural patterns zu definieren, um den Architekturprozess effizienter zu gestalten. Hier sind aber noch viele Fragen zu klären, z. B. nach welchen Merkmalen und Anwendungsbereichen derartige patterns zu konzipieren sind, wie Kontextinformation mit eingebracht, wie tradeoffs dokumentiert und berücksichtigt werden.

Vielfach wird die Forderung erhoben: „System vor Funktion“. Damit wird gefordert, dass Systembasisfunktionen, wie Betriebssystem, Netzwerkmanagement oder Diagnosekommunikation, zusammengefasst als Standard Core bezeichnet, zuerst implementiert und verifiziert sein müssen, bevor die Applikationsfunktionen

realisiert werden. Diese Forderung ist zwar grundsätzlich richtig, darf aber nicht absolut gesetzt werden, da für die Erprobung von Fahrzeugfunktionen auch in frühen Reifephasen eine Reihe von Applikationen verfügbar sein müssen. Für die praktische Systementwicklung ist es erforderlich, die Funktionen mit Priorität zu entwickeln und bereit zu stellen, die für die jeweilige Reife- und Erprobungsphase des Fahrzeuges von Bedeutung sind. Dies setzt nun allerdings voraus, dass man eine klare Integrationsplanung mit entsprechenden Reifestufen aufsetzt und die funktionalen Zusammenhänge und Abhängigkeiten beschreiben und analysieren kann. Auch hier ist die enge Zusammenarbeit zwischen Systemarchitekten, Domänenarchitekten, Infrastrukturverantwortlichen und den Spezialisten für die Verifikation notwendig.

Eine gute Integrationsfähigkeit eines Systems setzt damit eine Reihe von Grundanforderungen an das Systemdesign voraus:

- Klare Festlegung der Prämissen der Systemarchitektur (z.B. Technologieentscheidungen)
- Möglichst vollständige Anforderungsanalyse
- Funktionsnetzwerkanalyse und -modellierung
- Strukturierte, modellbasierte Entwicklungsmethodik
- Standardisierung der Systembasisfunktionen (Standard Core)
- Offene Systemarchitektur
- Modulare Hardware- und Software-Architektur
- Schnittstellenstandardisierung
- Hardware-abstrahierenden Middlewareschicht

Für den Verifikationsprozess sind die Umfänge der Testaktivitäten im Rahmen der Modul- und Komponententests, im Rahmen der Teilsystemplätze und schließlich der Systemtests am statischen und dynamischen Laborfahrzeug (statischer oder fahrfähiger Erprobungsträger der aktuellen Hardware- oder Softwarestände) sorgfältig inhaltlich und terminlich aufeinander abzustimmen. Fehlende, unvollständige oder zeitlich überlappende Testaktivitäten erhöhen den Aufwand bei Fehleranalyse und Systemintegration erheblich und können zur völligen Überforderung der verfügbaren Ressourcen oder zur massiven Einschränkung der Testtiefe führen.

Die Tendenz in immer schnellerer Folge inkrementelle Systemverbesserungen, durch SW-Builds teilweise im 2 Tages Rhythmus zu realisieren, führt entweder zu einer drastischen Aufblähung der Test- und Update-Aufwände, da der relative Overhead-Anteil für Test und Update gleich bleibt, oder zu einer unvermeidbaren Reduktion von Testtiefe und –breite. Inkrementelle Entwicklungsschritte sind in der modernen Software-Entwicklung zwar unvermeidbar, die daraus resultierenden Risiken müssen aber durch Phasen mit ausreichender Testzeit und klar definierten Integrationsständen wieder aufgefangen werden. Alle praktischen Erfahrungen zeigen, dass das Risiko von Kompatibilitätsverletzungen durch Softwareänderungen umso geringer ist je klarer und präziser das geforderte Schnittstellenverhalten beschrieben ist. Hier bieten sich in der Zukunft noch erhebliche Spielräume insbesondere auch das dynamische Verhalten an den Schnittstellen präzise zu beschreiben, Ein Entwicklungsprozess der den Entwurf deterministischer Systeme zum Ziel hat, kann hier die richtige Basis legen.

.

#### **4. Zusammenfassung**

Systemintegration ist mehr als das Zusammenfügen von Komponenten zu einem funktionsfähigen Gesamtsystem. Systemintegration muss in Zukunft in enger Verzahnung zwischen Technologienentwicklung von Infrastrukturelementen Architekturkonzeption und Prozessgestaltung erfolgen. Systemintegration ist nicht ein Prozess der nur jeweils für ein bestimmtes Fahrzeugprojekt zu leisten ist, sondern ein Prozess der in den Gesamtrahmen einer längerfristigen, produktlinienübergreifenden Architekturstrategie eingebettet sein muss und mit der Technologievorentwicklung beginnt.

# Zukünftige E/E-Systemarchitekturen

## Zusammenfassung



Architektur ist  
Enabler für:

Innovation,  
Qualität und  
Integration

Reduktion der Heterogenität, Modularisierung und stärkere Hierarchisierung sowie offene Architekturen sind Schlüsselfaktoren zukünftiger Architekturen.

*Bild 12: Systemarchitektur zur Darstellung von Innovation, Einsatz moderner Technologie Enabler und Befähigung des Komplexitätsmanagement*

Die Entwicklungs- und Verifikationsprozesse dürfen sich nicht verselbstständigen sondern müssen ständig in enger Verkopplung mit der Technologieentwicklung weiter fortgeschrieben werden. Hier sind in den nächsten Jahren insbesondere starke Verlagerungen von aufwändiger hardwarebasierter Absicherung Richtung modellbasierter Absicherung zu erwarten.

Systemintegration ist die zentrale Zukunftsaufgabe der Automobilindustrie, die es ermöglicht den Zielkonflikt zwischen hoher Funktionalität, Kosten und Qualität zu bewältigen.

Literaturhinweise:

- [1] [www.autosar.org](http://www.autosar.org)  
Offizielle website der AUTOSAR development partnership
- [2] H. Heinecke, K.-P. Schnelle, H. Fennel, J. Bortolazzi, L. Lundh, J. Leflour, J.-L. Maté, K. Nishikawa, T. Scharnhorst, AUTomotive Open System ARchitecture - An industry-wide initiative to manage the complexity of emerging Automotive E/E-Architectures Convergence 2004, International Congress on Transportation Electronics, Detroit, 2004, p. 325ff

- [3] T. Scharnhorst, H. Heinecke, K.-P. Schnelle, H. Fennel, J. Bortolazzi, L. Lundh, J. Leflour, P. Heitkämper, j.-L. Maté, K. Nishikawa, AUTOSAR – Challenges and achievements  
Electronic Systems for Vehicles 2005, VDI Congress, Baden Baden, 2005, p. 395ff
- [4] Chr. Salzmann, M. Rudorfer, M. Thiede, T. Ochs, P. Hoser, M. Mössmer, H. Heinecke, Erfahrungen mit der technischen Anwendung einer AUTOSAR Runtime Environment – Migrationsstrategien in die AUTOSAR Architektur, VDI Berichte 1907, 2005, p.465ff
- [5] A. Schedl, J. Berwanger, M. Peller, BMW- first series cars with FlexRay in 2006, automotive electronics + systems, special edition FleyRay, Hanser, 2005, p. 6ff
- [6] K.Gresser und H.Heinecke,  
OSEK/VDX im BMW Standard Core  
In Car computing, Praxis Profiline, Vogel – Verlag, Würzburg, 2001, p. 32-35