



BMBF-Verbundprojekt: VISION  
Förderkennzeichen: 01M3078  
Projektlaufzeit: 01.05.2006 bis 31.08.2009

### Schlussbericht

## **Verteilte integrierte Systeme und Netzwerkarchitekturen für die Applikationsdomänen Automobil und Mobilkommunikation (VISION)**

Version: 1.0  
Erstelldatum: 11.3.2010  
Autoren: Oliver Bringmann, Matthias Krause, Alexander Viehl,  
Jochen Zimmermann (FZI Forschungszentrum Informatik)  
Jens Stellmacher (Cadence Design Systems GmbH)  
Ulrich Nageldinger (Infineon Technologies AG)  
Joachim Gerlach, Peter Jores (Robert Bosch GmbH)  
Dieter Treytnar (edacentrum GmbH)  
Zuwendungsempfänger: FZI Forschungszentrum Informatik (FZI)  
Cadence Design Systems GmbH (CDNS)  
Infineon Technologies AG (IFX)  
Robert Bosch GmbH (RB)  
Ansprechpartner: Peter Jores (RB)  
Tel. +49 7121 35-2982  
Fax +49 711 811-5142982  
Peter.Jores@de.bosch.com

*Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01M3078 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.*

© Copyright 2006-2009 by FZI Forschungszentrum Informatik, Cadence Design Systems GmbH, Infineon Technologies AG und Robert Bosch GmbH.

---

*This document and the information contained herein may not be copied, used or disclosed in whole or in part outside of the consortium except with prior written permission of the listed partners above.*

# 1 Kurzfassung

VISION wurde für 3 Jahre unter dem Förderkennzeichen 01M3078 im Förderprogramm Ekompas durch das BMBF gefördert. An dem Projekt beteiligten sich namhafte Vertreter aus Industrie und Forschung an fünf deutschen Standorten.

- FZI Forschungszentrum Informatik (FZI), Karlsruhe
- Cadence Design Systems GmbH (CDNS), München
- Infineon Technologies AG (IFX), Neubiberg mit  
Eberhard Karls Universität Tübingen (UTU-TI) und  
Universität der Bundeswehr München (UniBW)
- Robert Bosch GmbH (RB), Reutlingen mit  
OFFIS e.V. - Institut für Informatik, Oldenburg (OFFIS)

Im Automobilbereich als auch im Bereich der Mobilkommunikation ist die Fähigkeit, eine Vielzahl von neuen Funktionalitäten in einem verteilten Systemszenario zu integrieren, eine Voraussetzung geworden, um im Markt zu bestehen. Und dies muss natürlich ohne Senkung der Produktivität und mit hohen Sicherheits- und Qualitätsanforderungen bei gleichzeitiger Betrachtung der wirtschaftlichen Aspekte erreicht werden. Die Erforschung einer domänenübergreifenden Entwurfsmethodik für verteilte mikroelektronische Systeme unter Berücksichtigung komplexer Umgebungsbedingungen und effizienter applikationsspezifischer Entwurfsprozesse war daher das Ziel von VISION.

Das Projekt VISION adressierte den Entwurf von vernetzten mikroelektronischen Systemen in den Bereichen Automobilelektronik und Mobilkommunikation und stellt erstmalig eine ganzheitliche Sicht auf vernetzte Systeme bereits in frühen Entwurfsphasen bereit, die es nun erlaubt, die aufgeführten neuen Anforderungen in den genannten Märkten erfolgreich zu beherrschen. Insbesondere das Ergebnis, die Systemintegration bereits zu Beginn des Entwurfsprozesses zu berücksichtigen, um Integrationsfehler bereits in der Spezifikationsphase aufdecken zu können, wird in der in Deutschland außerordentlich wichtigen Disziplin der Systemintegration zu einer starken Produktivitätssteigerung führen und die Fähigkeit zum methodischen Entwurf zukünftiger hochvernetzter mikroelektronischer Systeme deutlich verbessern. Da die Systemintegration im Automobil- und Telekommunikationssektor einen sehr hohen Anteil in der Wertschöpfungskette besitzt, sind direkte positive mikro- und makroökonomische Auswirkungen naheliegend.

Weitere Informationen zu VISION finden Sie auf der Projekt-Homepage [VISION].

---

## 2 Inhalt

<b>1</b>	<b>Kurzfassung</b>	<b>3</b>
<b>2</b>	<b>Inhalt</b>	<b>4</b>
<b>3</b>	<b>Ziele und Aufgaben</b>	<b>5</b>
3.1	<b>Ausgangssituation und Voraussetzungen, unter denen das Vorhaben durchgeführt wurde</b>	<b>5</b>
3.2	<b>Planung und Ablauf des Vorhabens</b>	<b>9</b>
3.3	<b>Wissenschaftlicher und technischer Stand, an den angeknüpft wurde</b>	<b>10</b>
3.4	<b>Zusammenarbeit mit anderen Stellen</b>	<b>17</b>
<b>4</b>	<b>Technische Ergebnisse</b>	<b>20</b>
4.1	<b>AP1: Topologien und Architekturen verteilter Systeme</b>	<b>20</b>
4.2	<b>AP2: Analyse und Bewertung der Eigenschaften verteilter Systeme</b>	<b>42</b>
4.3	<b>AP3: Verifikationsgestützte Systemintegration und –implementierung</b>	<b>52</b>
4.4	<b>AP4: Applikationen und Designflow-Integration</b>	<b>60</b>
4.5	<b>Voraussichtlicher Nutzen</b>	<b>64</b>
4.6	<b>Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen</b>	<b>70</b>
<b>5</b>	<b>Veröffentlichungen und Patente</b>	<b>71</b>
<b>6</b>	<b>Quellenverzeichnis</b>	<b>75</b>
<b>7</b>	<b>Tabellenverzeichnis</b>	<b>80</b>
<b>8</b>	<b>Abbildungsverzeichnis</b>	<b>81</b>

---

## 3 Ziele und Aufgaben

### 3.1 Ausgangssituation und Voraussetzungen, unter denen das Vorhaben durchgeführt wurde

Der zunehmende Trend zu einer allgegenwärtigen Unterstützung des Menschen durch intelligente Systeme hat fundamentale Auswirkungen auf die Welt zu Beginn des 21. Jahrhunderts. Die treibenden Kräfte sind Interaktivität, Mobilität sowie individuelles Bedürfnis nach Sicherheit, Schutz, Komfort und Gesundheit in einer hochgradig vernetzten Welt. Diese Veränderung spiegelt sich in der Vernetzung individueller Rechenleistung über unterschiedliche Informations- und Kommunikationstechnologien wider. Während einerseits Netzwerk und Protokolle nahezu unsichtbar für den Benutzer sind, haben sie andererseits signifikante Auswirkungen sowohl auf die Leistungsfähigkeit als auch auf den Entwurf verteilter mikroelektronischer Systeme und erweisen sich zunehmend als äußerst kritische und nur schwer beherrschbare Entwurfsprobleme.

Intelligente eingebettete Systeme sowohl in der Automobil- als auch in der Telekommunikationstechnik sind zunehmend durch einen steigenden Vernetzungsgrad geprägt. So wird in Zukunft neue Funktionalität weniger durch die Summe der Einzelkomponenten sondern durch deren gegenseitige Vernetzung realisiert. Neue Anwendungen werden daher zukünftig überwiegend aus der Vernetzung resultieren. So ist etwa denkbar, dass innerhalb eines vernetzten Systems eine neue Systemfunktion realisiert wird, ohne dass hierfür der Netzwerkstruktur neue Komponenten hinzuzufügen sind – die neue Systemfunktion also vollständig „aus der Vernetzung“ resultiert. Dieser Wandel im Produktbereich zwingt zunehmend auch zu einem Paradigmenwechsel im Entwurf. Der bestehende komponentenzentrierte Entwurf muss einer ganzheitlichen Sicht eines vernetzten eingebetteten Systems weichen, um frühzeitig die Auswirkungen der Vernetzung analysieren und bewerten zu können.

Die beschriebenen Veränderungen fanden zum Projektstartpunkt keine Berücksichtigung in Entwurfsmethodiken und -werkzeugen auf Systemebene. So fokussieren bisherige Ansätze auf Einzelsysteme und vernachlässigen den Einfluss der Integration von Systemen innerhalb ihrer hochvernetzten Umgebung. Etwa befindet sich bereits in heutigen Automobilen eine Vielzahl vernetzter Steuergeräte, die über ein heterogenes Netzwerk, bestehend aus unterschiedlichen Bussen und Protokollen (CAN, LIN, MOST, FlexRay etc.), miteinander verbunden sind. Ebenso bestehen moderne Mobiltelefone mittlerweile aus mehreren miteinander verbundenen Subsystemen sowie vielfältigen Peripheriekomponenten, wie Kameras und Memory-Card-Schnittstellen, die über heterogene Verbindungsstrukturen (I<sup>2</sup>C, I<sup>2</sup>S, I<sup>2</sup>RF etc.) kommunizieren. Die Kommunikationsarchitektur hat dabei signifikanten Einfluss auf die Funktionalität und das zeitliche Verhalten des gesamten Systems. Dennoch werden deren Einflüsse erst dann berücksichtigt, wenn alle Subsysteme vollständig in einem realen Prototyp integriert worden sind. Dies führt dazu, dass Integrationsfehler erst sehr spät aufgefunden werden. Bei der steigenden Komplexität entartet dies zunehmend zu einem „Trial-and-Error“-Integrationsprozess von vernetzten mikroelektronischen Systemen.

Abbildung 1 zeigt exemplarisch heutige und zukünftige automobilelektronische Systemfunktionen, deren wirtschaftlicher Entwurf und zuverlässige Integration in eine hochgradig vernetzte Systemarchitektur eine bisher ungelöste Herausforderung an neue Entwurfsmethoden darstellt.

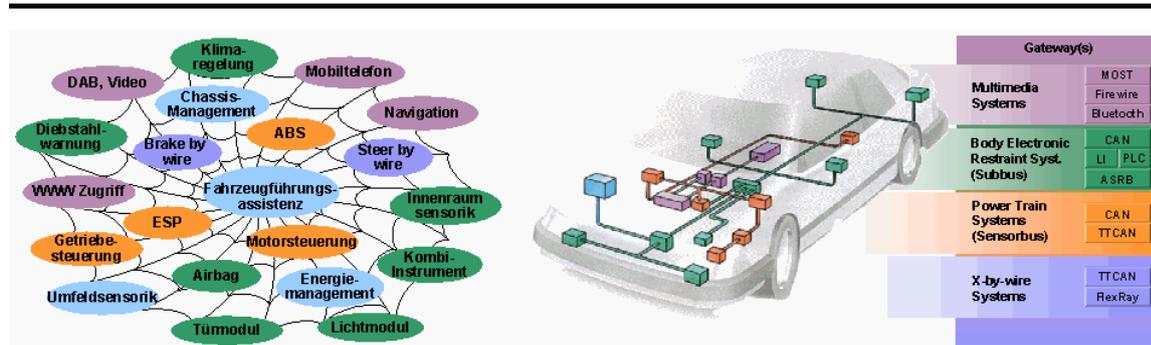


Abbildung 1: Vernetzte automobilelektronische Systemfunktionen

Projiziert auf die Designfähigkeit künftiger deutscher Schlüsselapplikationen lässt sich ableiten, dass mit verfügbaren Mitteln der Entwurf vernetzter Systeme, etwa x-by-wire Systeme im Bereich Automobil oder die Kopplung von Mobiltelefonen mit Einheiten zur Positionsbestimmung und komplexer mobiler Endgeräte im Bereich der Telekommunikation, mit wirtschaftlich vertretbarem Aufwand nicht mehr möglich sein wird. Diese Systeme zeichnen sich durch extrem hohe Anforderungen an Flexibilität, Zuverlässigkeit, Performanz, Echtzeit- und Fehlertoleranzverhalten aus. Andererseits besitzen derartige Systeme jedoch einen enormen volkswirtschaftlichen Nutzen und damit wirtschaftliche Relevanz für den Industriestandort Deutschland: Wie aktuelle Automobil-Studien belegen, sind ca. 60% aller Unfälle mit Todesfolge auf Seitenkollisionen zurückzuführen, die in den meisten Fällen aus einer Fehlreaktion des Fahrers (Schrecksekunde, Übersteuern des Fahrzeugs, etc.) resultieren [Quelle: RESIKO Report, GVD]. Durch intelligente Systeme wie x-by-wire können diese Fehlreaktionen erkannt und durch aktive Eingriffe des Systems in das Lenk- und Bremsverhalten des Fahrzeugs kompensiert werden. Dadurch ließe sich ein Großteil besagter Unfälle vermeiden oder deren Folgen für Leib und Leben minimieren.

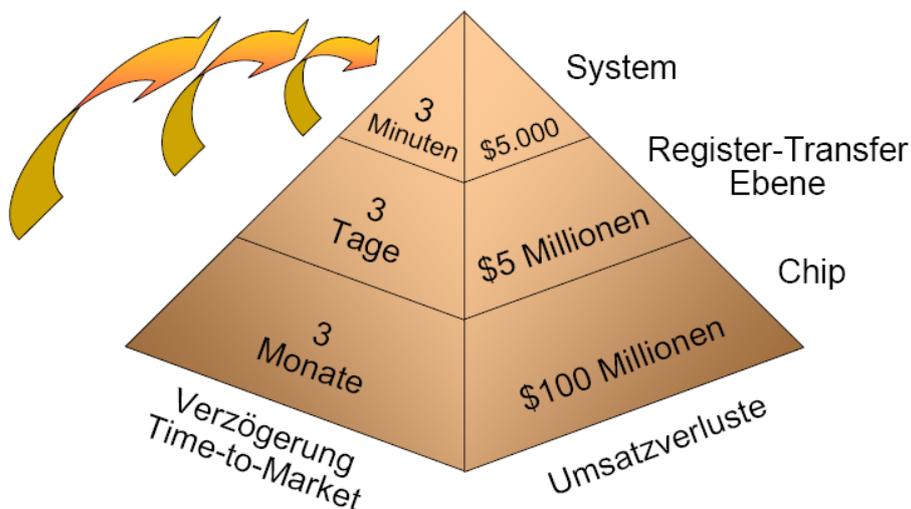
Im Telekommunikationsbereich werden zukünftig mobile Endgeräte zunehmend durch „artfremde“ Komponenten (also Funktionsmodulen, die nicht der eigentlichen Kommunikation dienen) erweitert und dadurch – neben einer Erhöhung von Mobilität und Komfort – neue Anwendungsbereiche erschlossen: Als Beispiel zu nennen sind hier etwa Mobiltelefone mit integrierter Gesundheitsüberwachung, die kontinuierlich eine Erfassung und Analyse der Körperfunktionen eines pflegebedürftigen Menschen vornehmen. Im Notfall setzt das Mobiltelefon automatisch einen Notruf mit Angabe der Positionskordinaten der Patientin/des Patienten ab und kann so Leben retten. Neben der besprochenen Entwicklung von Funktionalität bei Handys kommt als zweiter Aspekt eine erwartete Fragmentierung des Marktes hinzu, sodass es in Zukunft bei High-End Feature Phones dedizierte Geräte für z.B. Spiele, Video, Business, usw., geben wird. Diese Fragmentierung schlägt sich auch im Entwurfsprozess nieder, da Chiphersteller durch die Spezialisierung der Endgeräte noch schneller und gezielter auf Kundenwünsche eingehen müssen.

Eine notwendige Voraussetzung für Wettbewerbsvorteile sowohl im Automobilbereich als auch bei der Mobiltelefonie ist jedoch die Fähigkeit, eine Vielzahl von neuen Funktionalitäten ohne Senkung der Produktivität und von Sicherheits- und Qualitätsanforderungen in wirtschaftlicher Weise zu entwerfen und in einem verteilten Systemszenario zu integrieren. Diese Fähigkeit wird durch die folgenden zum Projektstart ungelösten Probleme und methodischen Lücken verhindert:

- Bestehende Systementwurfsmethoden konzentrieren sich auf den Entwurf von Einzelsystemen und vernachlässigen den Einfluss der Einbettung von Systemen innerhalb ihrer hochvernetzten Umgebung oberhalb der Ebene von SoCs/NoCs (Verbunde mikroelektronischer Subsysteme).

- Der Einfluss der Kommunikationsarchitektur auf das Gesamtsystem kann erst nach Integration aller Subsysteme in einen realen Prototypen analysiert werden, also erst am Ende des Systementwurfsablaufs.
- Als Folge erweist sich der Systemintegrationsprozess in vernetzten mikroelektronischen Systemen bei steigender Komplexität zunehmend zu einer „Trial-and-Error“-Vorgehensweise und führt zu einem verspäteten Auffinden von Integrationsfehlern, schlimmstenfalls im Endprodukt.

Abbildung 2 zeigt die Auswirkungen einer verspäteten Fehlerrückmeldung und den damit verbundenen Einbußen bezüglich Markteinführungszeit und Umsatz auf unterschiedlichen Abstraktionsebenen.



**Abbildung 2: Verspätete Markteinführung und Umsatzeinbußen aufgrund später Fehlerrückmeldung (Quelle: Integrated Communications Design May, 2001)**

Die Lösung dieser Probleme erfordert Synergien und Kooperationen, die über die Grenzen von Firmen und Forschungseinrichtungen hinausreichen und dabei sowohl Kompetenzen aus dem industriellen als auch aus dem akademischen Bereich bündeln. VISION knüpfte genau an dieser Stelle an und hatte die folgenden Ziele:

- Erfassung und Modellierung verteilter mikroelektronischer Systeme als ganzheitliche Einheit unter Berücksichtigung komplexer Umgebungsbedingungen des Systems, dadurch ganzheitliche Sicht auf vernetzte Systeme bereits in frühen Entwurfsphasen,
- methodische Analyse des Kommunikationsverhaltens und Identifikation einer optimierten Kommunikationstopologie und geeigneter Protokolle auf hoher Abstraktionsebene,
- Analyse und Simulation der Systemintegration bereits zu Beginn des Systementwurfsprozesses,
- Erforschung einer domänenübergreifenden Entwurfsmethodik für verteilte mikroelektronische Systeme und effiziente applikationsspezifische Entwurfsprozesse und
- Bereitstellung von „Enabler“-Lösungen, die den wirtschaftlichen Entwurf zukünftiger Systeme, wie x-by-wire in der Automobiltechnik sowie neue Generationen mobiler Kommunikationssysteme ermöglichen.

Diese Ziele unterstützen direkt die deutschen Schlüsselindustrien Automobil und Telekommunikation sowie die in Deutschland außerordentlich wichtige Disziplin der Systemintegration, die insbesondere im Automobil- und Telekommunikationssektor einen sehr hohen Anteil in der Wertschöpfungskette aufweist.

## Wissenschaftliche und technische Ziele des Vorhabens

Die im Rahmen des Projekts erforschten Methoden adressierten die Herausforderungen im Entwurf *verteilter* mikroelektronischer Systeme oberhalb der Ebene von SoCs und NoCs (Systemverbunde aus vernetzten Subsystemen).

Schwerpunkte bildeten dabei neue Spezifikationsmethoden für verteilte Systeme, Methoden zur automatisierten Ableitung von optimierten Verbindungstopologien für gegebene Applikationen sowie die Erforschung von Verfahren zur Performanz- und Kommunikationsanalyse für eine automatisierte Dimensionierung und Parametrisierung der Verbindungsstruktur. Weitere Schwerpunkte lieferten die Berücksichtigung von Randbedingungen der Systemumgebung in der Modellierungsphase, die automatisierte Ableitung virtueller Prototypen aus abstrakten Systemmodellen sowie neue Ansätze zur Verifikation von Modellen über Domänen- und Abstraktionsgrenzen hinweg. Lösungsansätze für die genannten Fragestellungen werden zukünftig die Integration von Systemen in ihre vernetzte Umgebung deutlich erleichtern und so den Prozess des „System Engineering“ wesentlich vereinfachen. Die nachfolgende Tabelle 1 fasst die wesentlichen von VISION adressierten Problemfelder zusammen und stellt konkrete Auswirkungen des Projekts auf die Entwurfbarkeit verteilter mikroelektronischer Systeme dar.

Problemfelder	ohne VISION		Ergebnisse von VISION	
	Entwurfsprozess	Auswirkungen	Entwurfsprozess	Auswirkungen
<b>Modellierung verteilter Systeme</b>	Keine vollständige Modellierung verteilter Systeme möglich, sondern unabhängige Modellierung der Teilsysteme	Ganzheitliche Systemsicht erst im realen Prototyp verfügbar, dadurch sehr späte Systemintegration	Ganzheitlicher Modellierungsansatz für verteilte Systeme und deren Umgebung	Modellbasierte Systemintegration bereits in früher Entwurfsphase
<b>Festlegung der Netzwerkstruktur</b>	Manuelle Bestimmung der Verbindungstopologie und manuelle Festlegung der Freiheitsgrade	Suboptimale Ausschöpfung des Optimierungspotenzials und zeitaufwändige Netzwerkoptimierung	Automatisierte Verfahren zur Bestimmung einer optimierten Netzwerkstruktur	Schnelle und sichere Optimierung der Netzwerkstruktur sowie Vermeidung zeitaufwändiger Entwurfsiterationen
<b>Bewertung des Gesamtsystemverhaltens</b>	Bewertung des Gesamtsystems erst mit realen Prototypen möglich	Auswirkungen des Verbindungsnetzes kann erst nach der realen Systemintegration analysiert werden	Modellbasierte Bewertung verteilter mikroelektronischer Systeme	Automatisierter Nachweis, dass das Modell gegebene Systemanforderungen erfüllt
<b>Verifikation der Systemintegration</b>	Verifikation des Gesamtsystems erst am realen Prototypen möglich	Sehr späte Aufdeckung von Integrations- und Interaktionsfehlern	Verifikation des automatisch generierten virtuellen Prototypen gegen das Systemmodell	Frühzeitige Aufdeckung von Integrationsfehlern ohne Verwendung eines realen Prototypen
<b>Einbettung in den Implementierungsprozess</b>	Keine durchgängigen Implementierungsprozesse für verteilte Systemlösungen verfügbar	Isolierte Implementierungsprozesse für Teilkomponenten, keine Erfassung von Umgebungseigenschaften	Durchgängiger Verfeinerungs- und Implementierungsprozess mit ganzheitlicher Systemsicht	Erfassung von Interaktions- und Umgebungseigenschaften, dadurch verbesserte Ausschöpfung des Optimierungspotenzials

Tabelle 1: Erwartete Ergebnisse von VISION hinsichtlich der Entwurfbarkeit

---

## 3.2 Planung und Ablauf des Vorhabens

Die Struktur des Projektes ergab sich unmittelbar aus den adressierten Defiziten und den daraus abgeleiteten wissenschaftlichen/technischen Ziele. Die folgenden Schwerpunktbereiche bildeten daher unmittelbar die Arbeitspaketstruktur des Projekts:

- Bestehende Methoden konzentrierten sich auf die Spezifikation und den Entwurf von (in sich komplexen) Einzelsystemen (etwa SoCs, NoCs etc.). Deren Integration in eine verteilte Systemlösung erfolgt in einem separaten Schritt in einer späten Phase des Entwurfsablaufs. Ebenso erfolgt im Entwurfsprozess keine strukturierte Modellierung der komplexen Anforderungen und Randbedingungen der Systemumgebung (des einbettenden Systems „Umwelt“) und übergeordneter Einflussgrößen, etwa aus dem Marketing. Im Rahmen der Projektarbeiten wurden Modellierungsansätze untersucht, die eine ganzheitliche Sicht auf die vernetzten Systeme bereits in frühen Entwurfsphasen unterstützen und so die Grundlage für Verfahren zur automatisierten Determinierung einer optimierten Kommunikationstopologie bilden. Eine Bearbeitung der genannten Fragestellung erfolgte schwerpunktmäßig in Arbeitspaket 1 („Topologien und Architekturen verteilter Systeme“).
- Zum Zeitpunkt des Projektstarts war das Vorgehen, den Einfluss der Kommunikationsarchitektur auf das Gesamtsystemverhalten erst nach der Integration aller Subsysteme in einem realen Prototyp, also erst am Ende des Systementwurfsablaufs, zu analysieren. Weiter fehlte eine methodische Unterstützung bei der Festlegung und Analyse der Kommunikationsstruktur. Im Rahmen der Projektarbeiten wurden Ansätze zur methodischen Analyse des Kommunikationsverhaltens verteilter mikroelektronischer Systeme erforscht, die eine Bewertung von Kommunikationstopologien in einer frühen Phase des Systementwurfsprozesses unterstützen. Diese bildeten den Schwerpunkt von Arbeitspaket 2 („Analyse und Bewertung der Eigenschaften verteilter Systeme“).
- Als Folge des im vorigen Punkt beschriebenen Defizits einer späten Berücksichtigung der Systemintegration entartet der Systemintegrationsprozess in vernetzten mikroelektronischen Systemen bei steigender Komplexität zunehmend zu einem „Trial-and-Error“-Prozess und führt so zu einer verspäteten und kostenaufwändigen Auffindung von Integrationsfehlern. Im Rahmen der Projektarbeiten wurden Verfahren erforscht, die eine Verifikation von Implementierungsmodellen der beteiligten Subsysteme im abstrakten verteilten Gesamtsystemmodell ermöglichen, damit Integrationsfehler frühzeitig aufdecken und so die bestehende Verifikationslücke schließen. Aus dieser Fragestellung leitete sich Arbeitspaket 3 („Verifikationsgestützte Systemintegration und -implementierung“) ab.
- Die zum Beginn des Projektes verfügbaren Entwurfsabläufe waren nicht in der Lage, die vielfältigen Interaktionen zwischen Kommunikationsnetzwerken und Netzwerkkomponenten in einem hochgradig verteilten Szenario (frühzeitig) zu erfassen und einem strukturierten Entwurfsprozess zuzuführen. Die im Rahmen der Projektarbeiten erforschten Methoden zur ganzheitlichen Erfassung verteilter Szenarien wurden in einen Entwurfsablauf integriert, der unter Ausnutzung der konkreten Kenntnis von Applikation und Anforderungen hochgradig anwendbare Designflows für verteilte mikroelektronische Systeme der deutschen Schlüsseltechnologien Automobil und Mobilkommunikation liefert. Diese bildeten den Schwerpunkt von Arbeitspaket 4 („Applikationen und Designflow-Integration“).

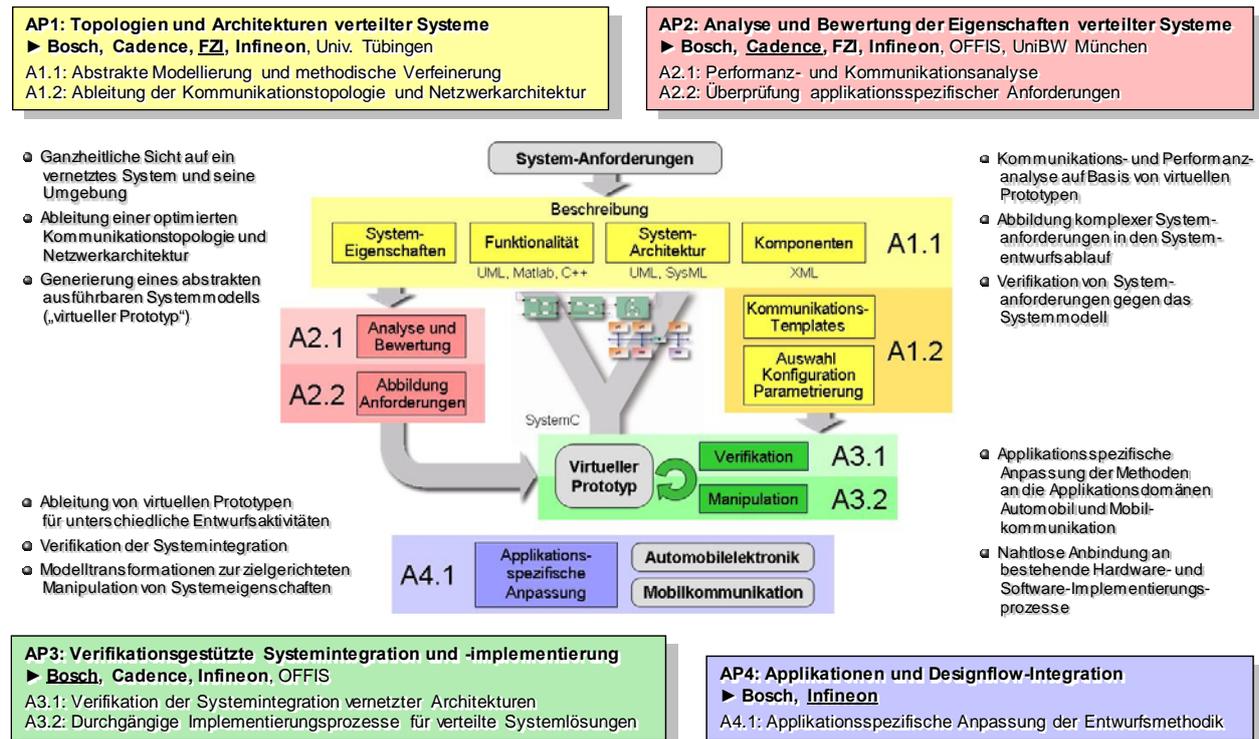


Abbildung 3: Projektstruktur und Arbeitspakete

Abbildung 3 zeigt die Projektstruktur von VISION und wie die 4 Arbeitspakete ineinander greifen.

### 3.3 Wissenschaftlicher und technischer Stand, an den angeknüpft wurde

Bis zum heutigen Zeitpunkt wurden in verschiedenen Bereichen der Informatik große Forschungsanstrengungen in Bezug auf verteilte Systeme unternommen. Etwa betrachtet der Bereich der Computer-Netzwerke eine große Spanne von Applikationen, die typischerweise zur Entwurfszeit des Systems noch nicht bekannt sind. Umgekehrt konzentriert sich der Bereich der verteilten Programmierung auf die Verteilung und Parallelisierung von Software auf einem in der Regel homogenen Netzwerk von Verarbeitungseinheiten. Im Gegensatz zu den genannten Disziplinen adressierte VISION verteilte Systeme in applikationsspezifischer Weise, wobei sowohl die Kenntnis der zugrunde liegenden Applikation als auch die Kenntnisse der zugrunde liegenden heterogenen Kommunikationsstruktur und Netzwerkarchitektur ausgenutzt werden.

Der enorme Bedarf an verteilten Systemlösungen mit rapide ansteigenden Anforderungen steht im erheblichen Widerspruch zur existierenden EDA-Landschaft: Zum Projektstart verfügbare Entwurfswerkzeuge waren auf den Entwurf von SoCs zugeschnitten. Sie waren jedoch nicht in der Lage, die spezifischen Randbedingungen *verteilter* Systeme zu unterstützen. Zu diesen zählen der hochgradig heterogene Charakter verteilter Systeme, die Anwendung hochkomplexer Kommunikationsprotokolle sowie der Einsatz standardisierter Subsysteme. Ein weiteres wichtiges Beispiel liefert der „offene“ Charakter verteilter Systeme: Bei SoCs wird während der Systemintegration die (physikalische) Konfiguration des Systems fixiert und nachfolgend nicht mehr verändert. Im Gegensatz dazu müssen verteilte Systeme auch *nach* der Systemintegration in der Lage sein, Erweiterungen des Systems um neue Komponenten, den Austausch von Systemkomponenten oder die Integration neuer Protokolle flexibel zu unterstützen. Insbesondere existierten keine Werkzeuge, die eine hochgradig verzahnte Interaktion zwischen Verbindungsnetzwerk und standardisierten Systemkomponenten berücksichtigen und in einem ganzheitlichen Ansatz erfassen.

---

Zwar bieten Systementwurfsumgebungen grafische Beschreibungsmöglichkeiten zur Modellierung einzelner Systeme (z.B. CoWare ConvergenSC, Synopsys System Studio, AXYS MaxSim, Summit Design Vista, ProsiLog Magillem), jedoch fehlte bislang eine strukturierte Vorgehensweise zur Komposition komplexer verteilter Systeme. Insbesondere setzen diese Werkzeuge bei der Systembeschreibung typischerweise bereits auf konkrete Sprachen (z.B. SystemC) zur Verhaltensmodellierung auf, wodurch die höchste Abstraktion bereits festgelegt ist. Es fehlte eine umfassende Beschreibung, die beliebig hohe Abstraktion erlaubt und auch Schnittstellen zu Geschäftsbereichen außerhalb des Entwurfs bietet. So bilden zwar die vom Marketing definierten Anwendungsfälle (Use-Cases) die Grundlage für Entwurfsanforderungen, deren nahtlose Überführung war jedoch bislang nicht möglich. Der verbreitete Modellierungsansatz UML 1.x (seit 2005 auch 2.0) bot hierzu keine befriedigende Lösung, da dessen Schwerpunkt stark auf die Bedürfnisse der Software-Entwicklung ausgerichtet ist und Hardware/Software-Systeme sowie (verteilte) Architekturen nicht hinreichend unterstützt werden. Weiterhin bestanden auf hohen Abstraktionsebenen Lücken im Entwurfsablauf. So werden für bestimmte Aufgaben des Concept Engineering abstrakte Systemmodelle benötigt, die mühsam und fehleranfällig aus Spezifikationen auf Papier erzeugt werden müssen. Es existierte keine Beschreibungsmethodik, die eine automatisierte Generierung abstrakter Systemmodelle bereitstellt. Im Rahmen der Design Automation Conference (DAC'2004, DAC'2005), auf welcher „Electronic System Level Design“ als zentrale Problemstellung identifiziert wurde, wurde auf dieses wichtige Problem in verschiedenen Veranstaltungen hingewiesen. Der begleitende Workshop „System-Level-Design mit UML“ konnte zwar einerseits die Vorteile der UML aufzeigen, andererseits wurde aber verdeutlicht, dass Spracherweiterungen sowie eine automatische Generierung von ausführbaren Systemmodellen eine notwendige Voraussetzung für den Einsatz der UML als Systemmodellierungssprache ist. Zwar sind erste UML-Profile für den SoC-Entwurf Gegenstand aktueller Entwicklungsaktivitäten, allerdings wurden methodische Ansätze zur Modellierung und Verfeinerung der Kommunikationsarchitektur bislang nur unzureichend adressiert, ebenso ist eine Anbindung an den Implementierungsprozess nur punktuell für eine Untermenge von Verhaltensdiagrammen verfügbar. Eine weitere wichtige Entwicklung in diesem Themengebiet stellt der SPIRIT-Standard für IP-Metabeschreibungen dar. Eine SPIRIT-Beschreibung definiert die Schnittstellen eines IP-Modules zur besseren Einbindung desselben in ein größeres System. Auf Implementierungsebene stellt dieser Ansatz ein Gegenstück zu einer UML-basierten Spezifikationsmethodik für verteilte Systeme dar, insbesondere unter dem Aspekt der automatischen Codegenerierung. Daher fand die SPIRIT-Beschreibung bei den Arbeiten zu diesem Thema besondere Berücksichtigung.

Beschreibungssprachen für eingebettete Systeme besaßen zu Projektbeginn entweder keine oder nur eine eingeschränkte Zeitsemantik. War die globale, ideale Zeit, die Sprachen wie VHDL und SystemC zugrunde liegt, ausreichend für den Entwurf räumlich begrenzter Hardware, stellte die Modellierung und der Entwurf vernetzter mikroelektronischer Systeme eine neue Herausforderungen dar. Zum einen muss es möglich sein, das reaktive Zeitverhalten verteilter Systeme mit Zeitschranken zu beschreiben, zum anderen ist es erforderlich, diese Informationen auf den unteren Entwurfsebenen umzusetzen. Es gibt zwar Arbeiten, die jeweils eine Entwurfsebene alleine betrachten, eine durchgängige Methodik für komplexe, verteilte Echtzeitsysteme existierte hingegen noch nicht. Insbesondere fehlten Verfahren, um das zeitliche Verhalten von Transaktionen mit hinreichender Genauigkeit zu spezifizieren.

Ansätze zur Analyse verteilter Systeme waren bislang sehr spezifisch auf eine Anwendungsdomäne zugeschnitten und deshalb nur äußerst eingeschränkt anwendbar. Aus dem Bereich der Echtzeit-Software existiert eine Reihe von Ansätzen zur Analyse nebenläufiger bzw. paralleler Softwaresysteme, die meist von einer Partitionierung der Prozesse in Tasks ausgehen, wobei jeder Task statisch bestimmbare Grenzen für die Ausführungszeit hat. Kommunikation findet jeweils nur am Anfang und Ende eines Prozesses statt. Das ursprüngliche Prozessverhalten wird durch einen azyklischen Task-Graphen dargestellt. Hierbei wird versucht, einen Ablaufplan (Schedule) zu finden, der die gegebenen Zeitschranken einhält.

---

Eine umfassende Systementwurfsmethodik muss sich auch mit der Verbindung zur Implementierung beschäftigen. Eine besondere Rolle spielen dabei Architekturmodelle auf Systemebene, die im Entwurfsprozess weiterverwendet werden. Solche sog. virtuellen Prototypen können als Grundlage für die Softwareentwicklung oder auch als Referenz für den Hardwareentwurf dienen. Das Problem hierbei besteht darin, dass je nach Anwendungsfall unterschiedliche Anforderungen an den virtuellen Prototypen gestellt werden. So gibt es hochperformante Lösungen, die auf die Software-Entwicklung abzielen. VaST Systems [Vast04] bietet z.B. hochperformante Prozessormodelle für bestimmte Prozessoren an, die mit Modellen von Peripheriekomponenten kombiniert werden können. Als Referenz für die Hardwareentwicklung sind diese Modelle wegen ihrer fehlenden Offenheit jedoch nicht geeignet. Andererseits bieten viele Umgebungen für den Systementwurf Lösungen auf Basis von SystemC [Bair03] an, z.B. CoWare ConvergenSC [Cowa04], oder verwenden proprietäre Methodiken, wie z.B. AXYS MaxSim [Axy04] [KrMa04]. Allen Lösungen gemeinsam ist der spezifische Anwendungsfall (meistens Software-Entwicklung). Mit den zum Projektbeginn existierenden Methoden konnten jedoch keine Modelle erzeugt werden, die auf unterschiedliche Anforderungen an virtuelle Prototypen angepasst werden können. Die genannten Systementwurfsumgebungen stellten auch den Stand der Technik für die (Performanz-) Analyse von Systemen durch Simulation dar. Auch hier gilt, dass zum einen verteilte Systemverbände nicht ausreichend unterstützt wurden, zum anderen musste die Systemarchitektur schon sehr genau definiert sein, um eine Simulation durchführen zu können. Die Anforderung, sehr früh (d.h. mit einer sehr abstrakten Beschreibung) bereits eine Aussage über die Systemleistung zu erhalten, erfüllten diese Umgebungen damit nicht. Eine provisorische Lösung stellt hier SystemC dar, das als Vehikel zur mühsamen, manuellen Erstellung von abstrakten Modellen dient, die dann Aussagen über die Systemleistung liefern.

Während im Bereich der Automobiltechnik die 80'er bzw. 90'er Jahre durch Komponenten mit zunehmendem Elektronikanteil bzw. durch den Aspekt der Systemintegration geprägt waren, definiert die komplexe Vernetzung elektronischer Steuergeräte im Automobil und in Zukunft die zunehmende Vernetzung von Automobilen und deren Umgebung die treibenden Herausforderungen der Automobildomäne [Quelle: Roadmap Automotive Electronics, Bosch]. Diese Entwicklung geht einher mit einem kontinuierlichen Anstieg der Zuverlässigkeits-, Performanz-, Echtzeit- und Fehlertoleranzanforderungen vernetzter Systeme. Der Trend zum hochgradig vernetzten Automobil wird durch verfügbare Methoden und Werkzeuge bisher kaum unterstützt. In heutiger Zeit sind lediglich punktuelle Lösungsansätze für den Entwurf und die Überprüfung spezifischer Bussysteme verfügbar. Aktuelle Standardisierungsaktivitäten im Automobilbereich beziehen sich entweder auf die Definition oder Weiterentwicklung spezifischer Schnittstellennormen und Netzwerkarchitekturen für dedizierte Bustypen mit klar abgegrenzten Einsatzbereichen und Leistungsanforderungen oder konzentrieren sich, wie etwa im Beispiel von AUTOSAR [Auto 04], auf die Vereinheitlichung der Software-Architektur verteilter Automobil-Applikationen. Verfahren der in VISION adressierten Art, die auf höherer Abstraktionsebene ansetzen und dort ausgehend von einer Modellierung von System, Umgebung und Anforderungen den strukturierten Entwurf einer hochgradig zuverlässigen Netzwerkarchitektur unterstützen, fehlten zu Projektbeginn vollständig. Dabei liefert gerade der Zuverlässigkeitsaspekt eine Problemstellung, an der heutige Entwurfsmethoden und -werkzeuge scheitern: So wird aktuell etwa seitens der Automobilindustrie ein teilweiser Innovationsstop diskutiert, für den Fall, dass weitere Innovationen nur mit Zuverlässigkeitseinbußen realisierbar sind. Dies belegt, dass Zuverlässigkeitseinbußen im Automobilbereich zukünftig nicht oder nur begrenzt durch zusätzliche Funktionalität kompensiert werden können. Dieses Defizit veranlasst in jüngster Zeit alle Beteiligten (OEMs, Zulieferer, etc.) zur Durchführung aufwändiger Maßnahmen zur Qualitätssteigerung ihrer Produkte. Hierzu erfolgt in den Firmen eine konsequente Strukturierung von Entwicklungs- und Herstellungsprozessen. Diese adressieren aber lediglich *bestehende* Prozesse und agieren nach wie vor auf Subsystem-Ebene, sind also nicht in der Lage, komplexe, aus der Vernetzung resultierende Zusammenhänge in der in VISION adressierten Art und Weise zu

berücksichtigen. Die Herausforderung bestand also darin, die (konkurrierenden) Anforderungen Komplexität und Zuverlässigkeit vernetzter Systeme gleichzeitig zu beherrschen.

### **Profil und Kompetenz der Antragsteller**

Das Projektkonsortium bestand aus Industriepartnern (Bosch und Infineon), die neben ihren methodischen Forschungsaktivitäten eine Parametrisierung der Entwurfsmethodik für verteilte mikroelektronische Systeme auf die jeweiligen Applikationsdomänen durchführten. Die Beteiligung des EDA-Partners Cadence, mit ausgewiesener Expertise im Bereich der Modellierung, Analyse und Simulation von Systemen, sorgte für eine Integration der erforschten Methoden und Verfahren in künftige Entwurfswerkzeuge für verteilte mikroelektronische Systeme. Die akademischen Partner (FZI, OFFIS, UniBW München, Universität Tübingen) betrieben aktiven Technologietransfer aus der akademischen Welt in das Projekt und verbreiteten die Ergebnisse durch Publikationen, Ausbildung und Training.

### **Robert Bosch GmbH**

Bosch ist weltweit einer der größten Automobilzulieferer. Bekannte Systeme aus dem Hause Bosch sind etwa Motor- und Getriebesteuerungen, Anti-Blockiersystem (ABS), Airbag, das elektronische Stabilitätsprogramm (ESP) sowie Fahrzeugradio- und navigationssysteme. Die Entwicklung von innovativen automobilelektrischen Systemen ist ein Hauptarbeitsgebiet der Entwicklung bei Bosch, wobei langjährige Erfahrungen sowohl in der Schaltungstechnik als auch bei den EDA-Werkzeugen vorliegen. Bosch konnte seine Kompetenz in der erfolgreichen Entwicklung vieler Systeme und in vielen Projekten unter Beweis stellen: Im Rahmen der Projekte Anastasia+ und VALSE(-XT) wurde Know-how im Entwurf analoger Systeme/Mixed-Signal-Systeme sowie im Bereich der Systemvalidierung unter Einsatz innovativer Techniken aufgebaut und im industriellen Umfeld untersucht. Die in diesen Projekten erzielten Ergebnisse spiegeln in hohem Maße die spezifischen Anforderungen der deutschen Schlüsseldomäne Automobil wieder, deren Systeme durch einen hohen Anteil analoger Systemkomponenten sowie durch einen hohen Sicherheitsanspruch charakterisiert sind und lieferten damit einen Treiber für die Problemstellung von VISION. Im Projekt SpeAC wurden spezifikationsbasierte Verfahren zur Modellierung und zum Entwurf komplexer Hardware/Software-Systeme erforscht, die ebenfalls wichtige Vorarbeiten für VISION darstellten: In SpeAC wurde eine Systementwurfsmethodik aufgebaut, die die spezifischen Anforderungen der Applikationsdomäne Automobil bereits auf hoher Abstraktionsebene berücksichtigt und in einen anwendbaren Entwurfsprozess einbettet.

### **Cadence Design Systems GmbH**

Die Cadence Design Systems GmbH ist die deutsche Tochterfirma eines weltweit führenden Anbieters von Entwurfswerkzeugen und Dienstleistungen für die Mikroelektronikindustrie. Innerhalb des Gesamtunternehmens bildet die deutsche Tochter eine treibende Kraft sowohl im Bereich der Methodenentwicklung als auch im Dienstleistungssektor. Mit diesen Eigenschaften bietet Cadence nicht nur die Möglichkeit für eine enge Kooperation auf Projektebene und effektive Verwertung der Projektergebnisse, sondern kann auch bei firmen-, technologie- oder applikationsspezifischen Erweiterungen wesentliche Unterstützung leisten. Cadence erwartet insbesondere im Systembereich in den kommenden Jahren eine steigende Nachfrage und möchte sich daher aktiv an der Entwicklung neuer Techniken, Methoden und Werkzeuge beteiligen. Da die Durchführung des Projekts VISION innerhalb von Cadence im Umfeld einer Dienstleistungsgruppe in enger Kooperation mit Cadence R&D als Pilotstudie angesiedelt war, konnte auf umfassende Erfahrung nicht nur im Bereich der Anwendung von Werkzeugen sondern auch im Bereich deren Integration in industrielle Entwurfsabläufe und deren durch industrielle Anforderungen getriebene applikationsspezifische

---

Erweiterung zurückgegriffen werden. Dies hat sowohl den Projektpartnern als auch deren potentiellen Kunden zusätzlich die Möglichkeit geboten, bei der Einbindung und Anwendung der Projektergebnisse auf einen erfahrenen, weltweit agierenden Partner zurückgreifen zu können. Darüber hinaus besitzt Cadence als einer der EDA-Marktführer die Möglichkeit, die Ergebnisse des Projekts in neue Entwicklungsstandards umzusetzen. Cadence Deutschland engagiert sich bereits seit über 10 Jahren in diversen Forschungs- und Entwicklungskooperationen im Rahmen von Smart System-Engineering (SSE), Ekompas und MEDEA+ Projekten. Dabei konnte im Hinblick auf die vorliegende Projektthematik umfangreiches Know-how aufgebaut werden. Umgekehrt konnten die deutschen Projektpartner bereits frühzeitig von den fortschrittlichen kommerziellen Entwurfswerkzeugen profitieren, bevor diese auf dem globalen Markt verfügbar waren.

### **FZI Forschungszentrum Informatik**

Das FZI ist eine mittelständische, gemeinnützige Forschungseinrichtung des Landes Baden-Württemberg mit der zentralen Aufgabe, neueste Ergebnisse wissenschaftlicher Forschung in mittelständische und industrielle Unternehmen zu transferieren. Die Abteilung „Systementwurf in der Mikroelektronik“ bringt seit Jahren ihre Erfahrungen bei der Konzeption und Evaluierung von Methoden und Werkzeugen für den Entwurf mikroelektronischer und verteilter eingebetteter Systeme erfolgreich in neue Projekte ein. Der Schlüssel dazu liegt in der Bündelung vorhandener Kompetenzen aus den Bereichen Systemspezifikation, Architekturbewertung, Schaltungssynthese und Virtual Prototyping. Die Qualität der Forschungsergebnisse spiegelt sich in einer Vielzahl von Publikationen auf namhaften internationalen Konferenzen (z.B. DATE, DAC, CODES-ISSS) sowie durch deren Transfer in mittelständische und industrielle Unternehmen wieder. Das FZI verfügt über eine international anerkannte Expertise, die unter anderem in einer Vielzahl von Forschungsprojekten erworben wurde und systematisch erweitert wird. Hierunter fallen Projekte, wie z.B. SpeAC, URANOS, CHESS, VERDE und SANITAS, die im Rahmen nationaler und europäischer Forschungsprogramme, wie z.B. IKT2020, MEDEA/MEDEA+, ITEA2 und ARTEMIS gefördert werden. Darüber hinaus wird die Verbreitung von SystemC vom FZI durch Leitung der European SystemC User Group sowie über Schulungen vorangetrieben. Ferner besitzt das FZI eine hohe Branchenkompetenz im Automobilbereich, die bereichsübergreifend im Rahmen des Living Labs Automotive gebündelt wird.

### **Infineon Technologies AG**

Infineon bietet Halbleiter- und Systemlösungen für Anwendungen in der mobilen Kommunikation, für Sicherheitssysteme und Chipkarten und für Automobil- und Industrieelektronik an. Aufgrund der Historie in komplexen SoCs in den Bereichen Automotive, drahtgebundene Kommunikation, Chipkarten, und Mobiltelefonie kann Infineon auf langjährige Erfahrung im Bereich System Level Design sowie Einsatz von neuen Beschreibungsmethodiken zurückblicken. Neben internen Projekten arbeitet Infineon hierbei auch an innovativen und zukunftsorientierten Förderprojekten. So wurden in den Projekten VALSE bzw. VALSE-XT innovative Methoden zur Verifikation sowie Kompetenz in diesem Gebiet aufgebaut. Weiterhin gibt es Vorarbeiten und Kompetenz zum Thema virtuelle Prototypen; es wurde sogar eine proprietäre Methodik entwickelt und eingesetzt, sowie eine aktuelle Modellierungsmethodik, die auf SystemC basiert. Im Projekt IP<sup>2</sup> untersuchte Infineon neben dem Einsatz von rekonfigurierbaren Hardware-Komponenten in mobilen Anwendungen (Car-Infotainment-Systeme) auch Schnittstellen von einem allgemeinen System-Level Flow zu anwendungsorientierten Entwurfsmethodiken (im Falle von IP<sup>2</sup> für Mobilkommunikations-Anwendungen). Im Projekt SpeAC bearbeitete Infineon Methodiken für Architekturentwurf von SoCs und den plattformbasierten Entwurf, wobei bereits verschiedene Modellierungs- und Simulationsansätze untersucht wurden. Die Erfahrungen bei der Systemspezifikation konnten dadurch in das VISI-

ON Projekt mit eingebracht werden. Da sich die Arbeiten in VISION im Gegensatz zu SpeAC mit Systemverbunden statt mit einzelnen SoCs befassten, ergänzten sich die Aktivitäten von Infineon in beiden Projekten.

### **Unterauftragnehmer**

#### **Unterauftragnehmer OFFIS Oldenburg**

OFFIS ist ein anwendungsorientiertes Forschungs- und Entwicklungsinstitut, gegründet 1991 als eingetragener Verein. Aktuell beschäftigt OFFIS ca. 240 Mitarbeiter, von denen mehr als 130 als Forscher tätig sind. OFFIS ist in drei F&E Bereiche – Energie, Gesundheit und Verkehr - unterteilt, die sich jeweils einem industriellen Anwendungsfeld widmen. Das VISION-Projekt ist im Bereich Verkehr verankert. Die für VISION relevanten Forschungsschwerpunkte im Bereich Verkehr sind Electronic-System-Level Design- und Optimierungstechniken sowie Werkzeuge und Methoden für den Entwurf eingebetteter Systeme. OFFIS verfügt über international anerkannte Expertise auf diesen Gebieten, bestätigt durch viele internationale Publikationen und internationale Kooperationen. Insbesondere hat es Renommee im Bereich der Low-Power Optimierungstechniken und im objektorientierten System-Level Design erworben. Dieses Know-how wurde in diversen Forschungsprojekten erworben und systematisch erweitert. Zu diesen Projekte, gefördert im Rahmen nationale und internationaler Forschungsprogramme, gehören etwa PolyDyn(DFG), ICODES, POET, ANDRES (IST FP5/6) oder LEMOS, RapidMPSoC (MEDEA+). In jüngst angelaufenen Projekten, wie etwa COMPLEX (ICT FP7), SANITAS (BMBF) und CESAR (ARTEMIS) beschäftigt sich OFFIS u.a. mit mehrkriterieller Analyse und Optimierung komplexer eingebetteter Systeme und durchgängige Verifikationsverfahren sicherheitskritischer Systeme. Aktuelle Arbeiten zielen auf neuartige Architekturen (z.B. Multi-Core Plattformen) und auf Robustheitsfragen, die durch immer kleineren Strukturgrößen an Bedeutung gewinnen.

#### **Unterauftragnehmer Universität der Bundeswehr (UniBW) München**

Die Universität der Bundeswehr München wurde vor mehr als 30 Jahren zusammen mit ihrer Schwesteruniversität in Hamburg gegründet. Sie hat den Auftrag, für Offiziere und Offiziersanwärter der Bundeswehr und befreundeter Nationen eine auf den neuesten Erkenntnissen der Forschung basierende aktuelle Ausbildung zu garantieren und ihre Absolventen darüber hinaus auch weiterzubilden. Die für dieses Projekt wichtigen Vorarbeiten am Institut für Technische Informatik liegen auf den Gebieten der: Methoden zur Modellierung und zur Simulation diskreter Systeme, Leistungs- und Zuverlässigkeitsanalyse innovativer Rechnerstrukturen, Diagnose und Management in verteilten Systemen und dem Aufbau bzw. der Leistungsanalyse von Hochleistungsrechnern. Die für das Projekt VISION relevanten Vorarbeiten konzentrierten sich auf die Erweiterung von bekannten Modellen mit einer Annotierung durch Performancedaten. Dadurch kann man die Leistungsfähigkeit von Softwarekomponenten in einem Entwurfsprozess frühzeitig vorhersagen, indem UML-Diagramme annotiert und die zur Leistungsabschätzung notwendigen Modelle erzeugt werden. Die Erfahrungen zeigen, dass sich zur Leistungsabschätzung in verteilten Systemen die Analyse mit Warteschlangen (Queuing Networks) besonders bewährt. Zur Vorhersage der Leistungsfähigkeit von Software lieferten hierbei Lehmann, Lüthi und Xu einen wesentlichen Beitrag. Die neueste Forschung an der Universität der Bundeswehr demonstriert, dass für die Performance Prediction von Software die Queuing Netzwerke aus annotierten UML-Modellen automatisch generiert werden können. Weiterführende Forschungsarbeiten, die auf den Ergebnissen bei der Vorhersage von Softwareperformance beruhen, wurden in VISION für verteilte Systemverbunde (mit Hardware- und Software-Komponenten) und Netzwerkarchitekturen durchgeführt und exemplarisch erprobt.

**Unterauftragnehmer Universität Tübingen**

Die Universität Tübingen arbeitet bereits seit mehreren Jahren auf dem Gebiet der objektorientierten Modellierung von Hardware/Software-Systemen mit SystemC und Java. In dieser Zeit wurden einschlägige Arbeiten in den Bereichen Modellierung, Partitionierung, Co-Simulation und Synthese von Hardware/Software-Systemen mittels der Beschreibungssprache Java und SystemC gesammelt. Im Bereich Spezifikation erarbeitete die Universität Tübingen ein objektorientiertes Modellierungs-Paradigma, das auf dem Komponentenmodell JavaBeans aufsetzt. Dieses bietet die Möglichkeit, das Verhalten eines Systems graphisch zu modellieren und zu visualisieren. In weiteren Arbeiten wurden die Standardprogrammiersprachen C und C++ zur Spezifikation und Evaluierung von Systemen eingesetzt. Es entstehen so ausführbare Spezifikationen, welche eine statische und dynamische Analyse unterstützen. Darauf zugeschnittene Analysemethoden erlauben die weitgehend automatische Untersuchung des Systems. Im Bereich eingebetteter Systeme beschäftigt sich die Universität Tübingen mit der Untersuchung und Modellierung flexibler Zielarchitekturen für eingebettete Hardware/Software-Systeme einschließlich der Spezifikation, Partitionierung, Bewertung, Simulation und Verifikation. Die in diesen Bereichen aufgebaute Kompetenz konnte von der Universität Tübingen wirkungsvoll in VISION eingebracht werden.

### 3.4 Zusammenarbeit mit anderen Stellen

Während der Laufzeit von VISION wurden intensive Kooperationen zwischen den beteiligten Projektpartnern durchgeführt, da aufgrund der kleinen Größe des Projektkonsortiums Abstimmungen und Ergebnisdarstellungen in enger Zusammenarbeit erfolgten. Zusätzlich wurden Kooperationen mit anderen IKT2020-Projekten durchgeführt, von denen die wichtigsten mit deren für VISION relevanten Ergebnissen nachfolgend dargestellt sind.

#### URANOS:

In VISION wurde die in URANOS entwickelte Methodik zur Erfassung, Formalisierung und Verknüpfung von Anforderungen verwendet und dahingehend erweitert, dass auch Anforderungen über Chip-Grenzen hinweg spezifiziert, während der in VISION entwickelten Analysen verfolgt und die Ergebnisse der Anforderungsbewertung zurück zur Spezifikation geführt werden.

Die in VISION entwickelten Analysen zur Bewertung des Einflusses des Verbindungsnetzwerkes konnten dazu verwendet werden, die in URANOS entwickelte Methodik zur Risikobewertung nicht-funktionaler Eigenschaften um die Berücksichtigung des Einflusses der Kommunikationsinfrastruktur zu erweitern.

#### HERKULES:

Ein enger Austausch erfolgte mit dem Projekt HERKULES hinsichtlich der Integration und Nutzbarmachung von Techniken zur formalen Schnittstellenverifikation innerhalb einer Methodik zur Gesamtsystemverifikation. Weiter erfolgte eine Diskussion der entwickelten Ansätze im Hinblick auf die Anforderungen im Kontext der entstehenden Norm ISO-26262 für funktionale Sicherheit.

#### VeronA:

In Zusammenarbeit mit dem Projekt VeronA wurde die Anwendbarkeit der entwickelten Modellierungs- und Verifikationsmethodik auf Analog/Mixed-Signal Systeme vorangetrieben und so deren Einsatz in heterogenen Systemszenarien vorbereitet.

#### SyEnA:

Nachdem die Beschreibungsmethodik von VISION sich auf die Anforderungen von System-Level Entwurf konzentrierte, soll in SyEnA diese Methodik mit Daten für Analog/Mixed-Signal Komponenten erweitert werden. Dadurch kann die Generierungsmethodik auch für die automatische Erzeugung von Testcases angewandt werden. Darüber hinaus gab es einen intensiven Erfahrungsaustausch zum Top-Down Entwurf komplexer heterogener Systeme unter Berücksichtigung von AMS-Komponenten und zur Erstellung und Anwendung ausführbarer Modelle im Systementwurfsablauf.

#### SANITAS:

Die Generierungsmethodik, die in VISION erfolgreich für die Erzeugung von Entwurfsdaten entwickelt wurde, wird in SANITAS für die Generierung von Verifikationsumgebungen eingesetzt und weiterentwickelt. Dabei können die formalen Beschreibungen, die die Grundlage der Generierung darstellen, auch für die SANITAS Arbeiten verwendet werden.

Die Arbeiten und Ergebnisse von VISION fanden bei externen Projekten sowie bei Standardisierungsorganisationen wie SPIRIT und OSCI großen Anklang. Daher wurde schon frühzeitig begonnen, möglichst vielseitig externe Kooperationen aufzubauen. Die wichtigsten externen Kooperationen mit Projekten und Organisationen sind mit deren Ergebnis in Abbildung 4 dargestellt.

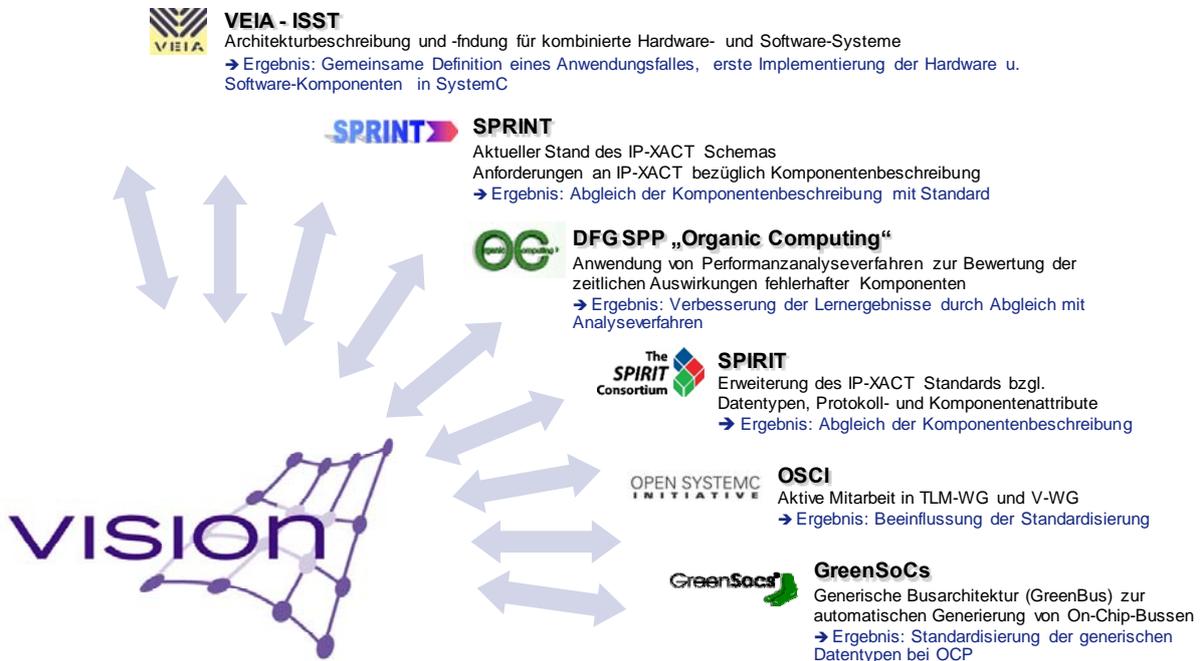


Abbildung 4: Kooperationen mit Projekten und Organisationen

## Highlights

### SPRINT / SPIRIT:

Sowohl im SPRINT-Projekt als auch im SPIRIT Konsortium fand ein ständiger Abgleich des VISION Datenmodells mit dem IP-XACT Standard ab. Dies zielte einerseits dazu ab, IP-XACT Beschreibungen vollständig abbilden zu können, andererseits konnten auch Konzepte aus dem VISION Datenmodell für IP-XACT zur Diskussion gestellt werden.

Die Zusammenarbeit mit externen Firmen wurde in VISION zur Ergebnisverbesserung groß geschrieben. Dabei ging es zu Beginn des Projektes um die Evaluierung von Werkzeugen, die für die Ergebniserarbeitung notwendig waren. Auf Ergebnisse des Projektes wie z.B. die Technologie für virtuelle Prototypen wurden Industriefirmen aufmerksam und suchten die Zusammenarbeit mit VISION. Die wichtigsten Kooperationen mit Firmen sind mit deren Ergebnis in Abbildung 5 dargestellt.



Abbildung 5: Kooperationen mit Firmen

Durch regelmäßige Veröffentlichungen auf Konferenzen und Workshops wurde über den Fortgang der Arbeiten informiert. Dabei erhielt insbesondere die Ergebnispräsentation auf der Design Automation Conference (DAC) im Juli 2009 großen Zuspruch.

## 4 Technische Ergebnisse

### 4.1 AP1: Topologien und Architekturen verteilter Systeme

Der Schlüssel für einen erfolgreichen Entwurf verteilter mikroelektronischer Systeme hängt maßgeblich von der Fähigkeit ab, diese Systeme ganzheitlich zu modellieren, um die durch die Vernetzung bedingten vielfältigen Abhängigkeiten beherrschen zu können. Die ganzheitliche Modellierung ist Grundvoraussetzung für eine frühe Systemintegration und unterstützt damit die Aufdeckung von Integrationsfehlern bereits beim Entwurfseinstieg. Weiterhin ist es wichtig, die Modellierungsmethodik nicht losgelöst von dem eigentlichen Implementierungsflow zu realisieren, da die entstehende Lücke im Designflow Ursache vieler manueller Verfeinerungsfehler ist und zu einer Senkung der Entwurfsproduktivität führt. Ebenso ermöglicht ein Modell des gesamten verteilten Systems eine automatisierte Bestimmung einer geeigneten Netzwerktopologie. Arbeitspaket 1 konzentrierte sich daher auf die Modellierung und automatisierte Generierung der Kommunikationstopologie und Netzwerkarchitektur für verteilte mikroelektronische Systeme. Es wurde eine abstrakte Beschreibungsmethodik zur Modellierung verteilter Systeme entwickelt und bereitgestellt, die eine methodische Verfeinerung des Modells erlaubt sowie eine automatisierte Ableitung einer geeigneten Netzwerkstruktur einschließlich geeigneter (ausführbarer) Modelle für eine weitere Prozessierung innerhalb des Entwurfsablaufs ermöglicht. Dies beinhaltet die Anwendung von Methoden für eine abstrakte modellbasierte Analyse des Systemverhaltens sowie die Bereitstellung und automatisierte Parametrisierung von generischen Kommunikationsschablonen.

Die Bereitstellung eines Ansatzes zur ganzheitlichen Modellierung verteilter mikroelektronischer Systeme wurde bislang nur punktuell für sehr spezifische Anwendungsbereiche gelöst. Ein allgemeingültiger Ansatz bietet somit erstmals die Möglichkeit, die Auswirkungen von gegenseitigen Abhängigkeiten innerhalb verteilter Systeme direkt beim Entwurf der Subsysteme aufzuzeigen und damit eine modellbasierte Systemintegration zu unterstützen, so dass Integrationsfehler frühzeitig aufgedeckt werden können. Des Weiteren unterstützt eine ganzheitliche Sicht auf ein verteiltes System die direkte Erfassung von Systemanforderungen aus dem Requirement-Engineering und erlaubt ferner Umgebungsmodelle systemweit zu berücksichtigen. Ein weiterer Aspekt betrifft die Möglichkeit, auf Basis des ganzheitlichen Systemmodells eine geeignete Kommunikationstopologie und Netzwerkarchitektur bereits in frühen Entwurfsphasen ableiten zu können und unterstützt somit einen nahtlosen Übergang von einem Anforderungs- über ein Spezifikations- hin zu einem Implementierungsmodell. Aufgrund der Notwendigkeit, heterogene Systeme modellieren zu können, die eine Vielzahl von Kommunikationsprotokollen verwenden und wiederum aus unterschiedlichen, heterogenen Subsystem bestehen, sind interdisziplinäre Auswirkungen zu erwarten. Die Mikroelektronik ist hierbei ein treibender Faktor, da integrationsbedingte Re-Designs viel kostenaufwändiger als in der Software-Technik sind.

#### **Aufgabe 1.1: Abstrakte Modellierung und methodische Verfeinerung**

##### **Beitrag 1.1.1: Semantik für eine UML-basierte Systembeschreibung und deren Einsatz zur Modellierung von mikroelektronischen Systemen**

Die stetig wachsende Komplexität und Modularität gegenwärtiger und zukünftiger Mikroelektronik erfordert die frühzeitige Möglichkeit einer Modellierung, die eine durchgehende Beschreibung kompletter verteilter mikroelektronischer Systeme in der Vielzahl ihrer Aspekte (Architektur, Implementierung, Kommunikation etc.) ermöglicht. Intelligente eingebettete Systeme sowohl in der Automobil- als auch in der Telekommunikationstechnik sind darüber hinaus zunehmend durch einen steigenden Vernetzungsgrad geprägt. So wird in Zukunft neue

Funktionalität weniger durch die Summe der Einzelkomponenten, sondern durch deren gegenseitige Vernetzung realisiert.

Dieser Wandel im Produktbereich zwingt zunehmend auch zu einem Paradigmenwechsel im Entwurf. Der bestehende komponentenzentrierte Entwurf muss einer ganzheitlichen Sicht eines vernetzten eingebetteten Systems weichen, um frühzeitig die Auswirkungen der Vernetzung analysieren und bewerten zu können.

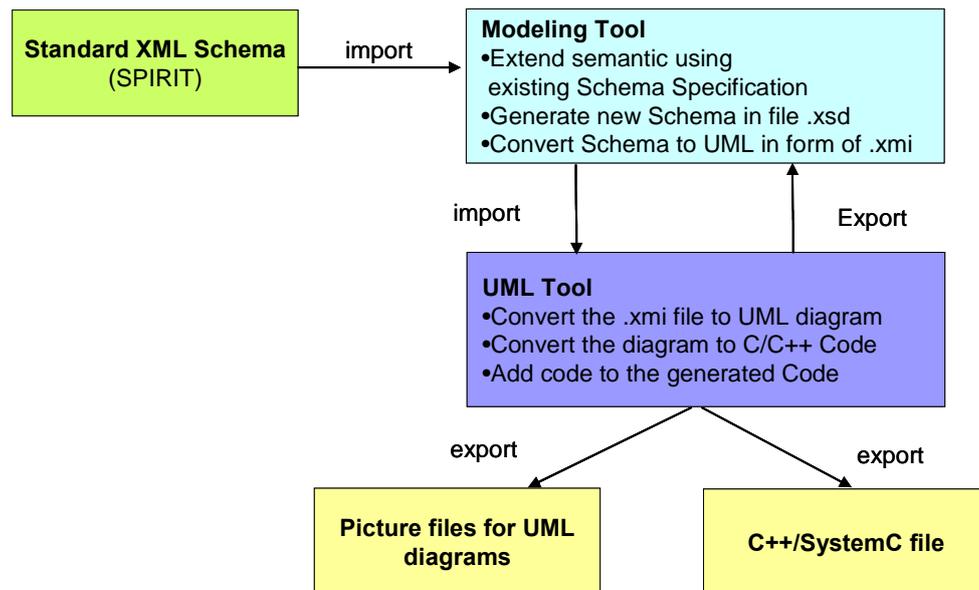
Die von Cadence durchgeführten Forschungsarbeiten zielten darauf ab, die Anforderungen für eine ganzheitliche Entwurfsmethodik zu erfassen und auf den verschiedenen Abstraktionsebenen exemplarisch anhand eines praxisrelevanten Beispiels zu untersuchen. Dies geschah in vier Schritten:

1. Identifikation einer geeigneten Semantik
2. Auswahl und Analyse von existierenden Modellen
3. Untersuchungen bzgl. einer geeigneten Darstellung der Semantik von mikroelektronischen Systemen
4. Erstellung eines Beispieldesigns zur Überprüfung der Vorgehensweise

Neben der Semantik als solcher, die die Repräsentation des Designs innerhalb eines Metamodells beschreibt, stand die Erfassung der relevanten Daten im Vordergrund. In Abstimmung mit den Partnern boten sich dabei die folgenden Sprachen und Formate an: UML und darauf aufbauend SYSML, XML und die in XML beschriebenen und standardisierten „Komponenten“ Beschreibungen SPIRIT und FIBEX sowie Matlab/Simulink als weit verbreitetes Modellierungstool.

Die Unified Modeling Language (UML) ist eine von der Object Management Group (OMG) entwickelte und standardisierte Sprache für die Modellierung von Software und anderen Systemen. Obwohl UML Systeme beschreibt, deckt der Standard nicht alle Praxisanforderungen ab. UML-Profile füllen diese Lücken, oder vielmehr sollen sie schließen, führen jedoch meist zu einem gehörigen Zusatzaufwand in der Beschreibung. SysML ist ein Subset von UML 2.0 und definiert gleichzeitig Erweiterungen, die notwendig sind, um die Anforderungen an UML von Seiten des Systems Engineering zu erfüllen. Er ist eine auf UML basierende standardisierte Sprache zur Spezifikation und Visualisierung objektorientierter Analyse- und Designmodelle. In unserer Arbeit haben wir UML am Anfang der Modellierung unseres Funktionsmodells eingesetzt, da es Diagramme mit oft benutzter Semantik im mikroelektronischen Bereich darstellen kann, z.B. FSM-, Aktivitäts- oder Sequenz-Diagramme. UML und SysML verwenden eine XML Metadata Interchange (XMI) Schnittstelleformat, um Modelle zwischen verschiedenen Werkzeugen auszutauschen, wobei die Umsetzung von UML zu XMI in den Arbeiten automatisch von einem Tool ausgeführt wurde.

Der Ablauf der untersuchten Methode zur Erstellung der Semantik und Modellierung mit Hilfe von UML, XML und verschiedenen Modellierungstools ist in Abbildung 6 dargestellt.



**Abbildung 6: Vorgeschlagene Anpassung von in XML und/oder UML beschriebenen Modelldarstellungen, um eine flexible und optimierte Repräsentation im Metamodell gewährleisten und gleichzeitig einen Codegenerator bedienen zu können.**

Um die Ausführbarkeit der entwickelten Methode und den Entwurfsablauf prüfen zu können, wurde ein Beispieldesign spezifiziert und für einen Teil, ein vereinfachtes sogenanntes *Brake-By-Wire* (BBW) Modell erstellt. Diese *Brake-By-Wire* System besteht aus unterschiedlichen Funktionsmodulen wie einem *Controller*, einem *Bus*, einem *Pedal* und mehreren *Sensoren* bzw. *Aktoren* zur Zustandsbestimmung bzw. Regelung.

Das BBW System ist ein vereinfachtes Modell sowohl auf der Seite der Funktionalität als auch auf der Seite der Kommunikation und Architektur. Um die Spezifikation in einer formale Form darzustellen, wurden verschiedene UML Diagramme wie das *Klassendiagramm*, das *Use-Case Diagram*, das *Aktivitätsdiagramm* und das *FSM State Chart Diagram* benutzt.

Da der Fokus unserer Arbeiten auf der Anwendbarkeit lag, sollte die verwendete Semantik so einfach wie möglich gestaltet sein. Nach unserer Sicht ist dies auch aufgrund der tatsächlichen Nutzung in der Industrie nur durch Unterstützung von UML, insbesondere SysML, bzw. MatLab/Simulink gewährleistet. Beide Vorgehensweisen sind werkzeuggestützt. Während die UML-Beschreibung einen tatsächlichen Standard darstellt, der seine größte Anwendung in der Entwicklung von Anwendungssoftware findet, stellt MatLab einen de facto Standard zur Verhaltensbeschreibung elektronischer Komponenten dar. Dieser wird innerhalb der Automobildomäne vor allem deshalb intensiv angewendet, weil er auch eine Beschreibung von analogem Verhalten durch Algorithmen ermöglicht sowie eine gemeinsame Modellierungs- und Simulationsoberfläche anbietet. Auch wenn wir beide Vorgehensweisen weiterverfolgen wollten, mussten wir uns bei den Arbeiten aus Aufwandsgründen für eine Vorgehensweise entscheiden, in diesem Fall war es SysML.

Um in der Entwicklung weiterzuvorschaufen zu können, muss die Möglichkeit einer Synthese des geschaffenen SysML Modells nach SystemC geschaffen werden. Da SystemC C++ die Nebenläufigkeit hinzufügt, kann nur so eine ernsthafte Modellierung von Nebenläufigkeit ermöglicht werden. Hierzu wurde ein Prototyp entwickelt, der es ermöglicht, das in einem XML Format exportierte SysML Model einzulesen und in einem eigenen Metamodell darzustellen. Zusätzlich besteht die Möglichkeit aus dem Metamodell ein SystemC Modell zu generieren und dessen Verhalten nach einer erfolgreichen Simulation mit einem SystemC-Simulator zu validieren.

---

Als Anwendungsfall für diese Entwicklungsumgebung steht das zuvor beschriebene Bremsensystem brake-by-wire zur Verfügung. Nach dem Einlesen des nach XMI exportierten SysML Modells kann eine Simulation gestartet werden, dessen Ergebnis wird automatisch gegen das ebenfalls aus dem SysML Modell exportierte Sequenzdiagramm validiert.

Das Ergebnis unserer Arbeiten ist eine automatisierte Generierung eines funktionalen C++ sowie eines SystemC Modells einschließlich einer Simulationsumgebung, die in der Architekturfindung und -bewertung eingesetzt werden kann.

### **Beitrag 1.1.2: Spezifikationsgetriebene Systemintegration verteilter mikroelektronischer Systeme mit UML/ SysML-basierter Systemmodellierung und automatischer Generierung virtueller Prototypen**

Zur Unterstützung einer frühen spezifikationsgetriebenen Modellierung verteilter mikroelektronischer Systeme wird ein Modellierungsansatz benötigt, der eine ganzheitliche Beschreibung des gesamten Systems erlaubt und die Modellierung der Zielarchitektur, der Software und der Kommunikation sowie des Zeitverhaltens auf unterschiedlichen Abstraktionsebenen umfasst. Um eine enge Verzahnung des Entwurfs der verteilten Hardwarearchitektur sowohl mit dem Requirements-Engineering als auch mit dem Entwurf der eingebetteten Software zu ermöglichen, wurde ein Spezifikationsansatz verfolgt, der auf einer globalen XML-basierten Spezifikationsdatenbasis beruht, auf der unterschiedliche Werkzeuge zur Modellierung (UML-Tools, XML-Editoren, Textprozessoren), Analyse und Modelltransformation sowie für das Requirementsmanagement operieren.

Die durchgeführten Forschungsaktivitäten des FZI zielten hierzu zunächst auf die Festlegung einer geeigneten Repräsentation von Prozessor- und Kommunikationsressourcen sowohl für deren direkten Verwendung in der XML-Spezifikationsdatenbasis als auch zur UML-basierten Spezifikation des Gesamtsystems mit der Möglichkeit zur anschließenden Transformation in eine Spezifikationsdatenbasis. Die XML-Beschreibung der Prozessor- und Kommunikationsressourcen basierte auf dem IP-XACT-Format (ehemals SPIRIT) und bot dadurch die notwendige Voraussetzung zur Generierung virtueller Prototypen in SystemC. Die Bereitstellung der Ressourcenbeschreibungen in UML sollte auf Basis eines UML-Profiles erfolgen, in dem unterschiedliche Ressourcen unter Verwendung von sogenannten „Tagged-Values“ und „Stereotypes“ gemäß der zugrunde liegenden IP-XACT-Beschreibung charakterisiert wurden. Um eine breite Akzeptanz des VISION-Ansatzes sicherzustellen, wurden kürzlich entwickelte UML-Profile (u.a. SysML, MARTE) hinsichtlich ihrer Anwendbarkeit zur Modellierung verteilter mikroelektronischer Systeme evaluiert. Hierbei hat sich gezeigt, dass keines der evaluierten Profile den gesetzten Anforderungen im VISION-Projekt genüge.

Deswegen wurden anhand der identifizierten Anforderungen an eine Modellierung verteilter mikroelektronischer Systeme in UML diese bereits bestehenden und teilweise standardisierter UML-Profile erweitert, um den in VISION adressierten Modellierungsansatz, der eine ganzheitliche Beschreibung eines verteilten mikroelektronischen Systems ermöglicht, zu realisieren. Ausgehend von einer Beschreibung mikroelektronischer Prozessor- und Verbindungsressourcen im standardisierten IP-XACT-Format wurde durch Integration dieser Komponenten in bestehende UML-Werkzeuge deren Verwendung als Komponentenbibliothek und dementsprechend deren Repräsentation in verschiedenen UML-Diagrammen ermöglicht. Dadurch entstand eine modellbasierte Verbindung zwischen unterschiedlichen Sichten der Systemintegration wie Hardware- bzw. Softwareentwicklung oder dem Requirements-Management, was z.B. die Abbildung von Funktionalität auf Komponenten und die Komposition dieser Komponenten zu einer Plattform auf hoher Abstraktionsebene unterstützte. Des Weiteren wurde hiermit der Ansatz einer Modelltransformation, sowohl hinführend zu einer modellbasierten Verfeinerung der Plattform als auch zu einer automatischen Generierung eines virtuellen Prototyps in SystemC, unterstützt. Die Festlegung der Anforderungen an die UML-Repräsentation und -Modellierung von mikroelektronischen Komponenten und deren

Detaillierungsgrad im UML-Modell (z.B. Ports, Register) erfolgte dabei in enger Abstimmung mit den Projektpartnern.

Basierend auf diesen Arbeiten wurde ein Designflow entwickelt, um mikroelektronische Prozessor- und Verbindungskomponenten, die aus IP-XACT-Beschreibungen als UML-Klassenbibliotheken importiert werden, in UML zu modellieren und zu einer Plattform komponieren zu können. Somit wurde die Darstellung einer gesamten Systemarchitektur ermöglicht. Durch Verwendung des UML-Profiles MARTE und zusätzlicher VISION-Profilerverweiterungen wurde die Semantik der zur Modellierung verwendeten Einheiten festgelegt und somit eine gemeinsame Modellierungsbasis geschaffen für eine Verbindung zwischen Hardware- und Softwareentwicklung. Das UML-Profil MARTE befand sich zu dieser Zeit bereits in fortgeschrittenem Stadium im Standardisierungsprozess durch die dafür zuständige OMG (Object Management Group), was nicht nur einen forschungsbezogenen sondern auch einen industriellen Einsatz des Profils zur Modellierung verteilter mikroelektronischer Systeme erlaubte (Abbildung 7).

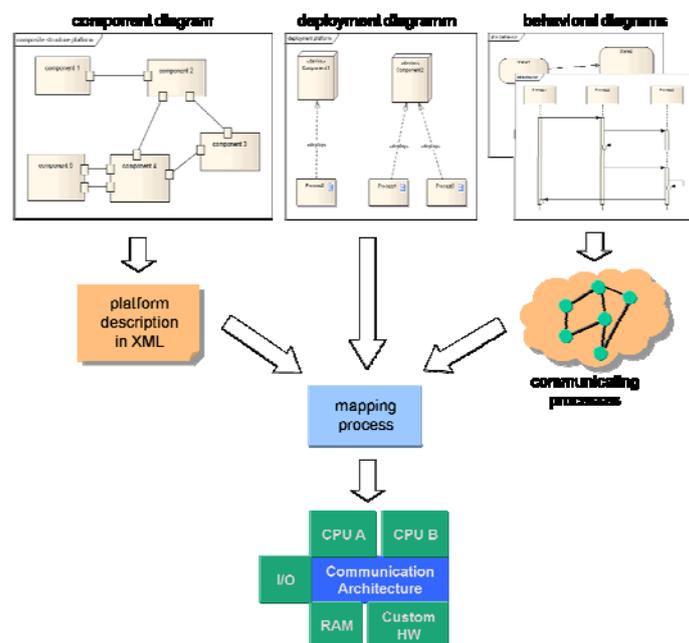


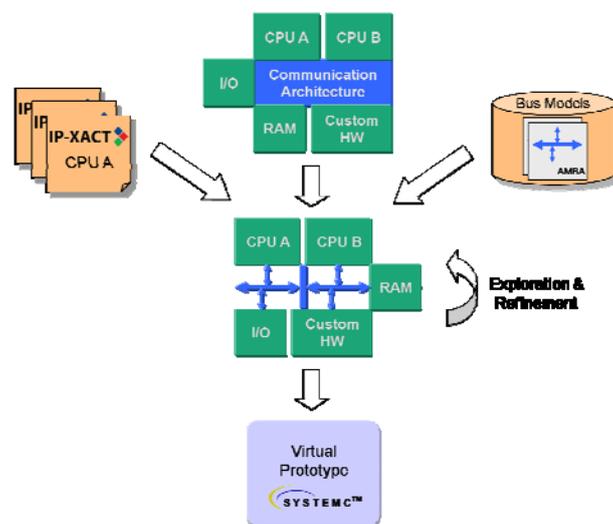
Abbildung 7: Modellierung und Abbildung von Plattformkomponenten

Um eine Template-basierte Systemkomposition, die im Arbeitspaket 1.2 adressiert wurde, zu ermöglichen, bot der entwickelte Designflow die Möglichkeit, die hierarchische Systemkomposition, in der modellierte Systeme in einer Schablone gekapselt und somit als Subsysteme in anderen UML-Plattformen wiederverwendet werden können.

Im weiteren Verlauf wurde ein Toolflow evaluiert, der einen Ablauf basierend auf der Entwicklungsumgebung *Eclipse* ermöglichen sollte. Die zuvor als Bibliotheken importierten Einzelkomponenten, die im IP-XACT-Format spezifiziert und durch Anwendung von Stereotypen des VISION-UML-Profiles nach UML transformiert wurden, werden in UML unter Berücksichtigung der Schnittstellenattribute zu einem Gesamtmodell kompositioniert. Den UML-Strukturdiagrammen des Eclipse-basierten Werkzeugs *Papyrus*, die die Systemkomposition in UML repräsentieren, liegt dabei das in *EMF* (Eclipse Modeling Framework) verwendete *Ecore*-Modell zugrunde. Durch eine Modell-zu-Modell-Transformation wurden bei einem *ARM/AMBA*-basierten Subsystem die Schnittstellenbeschreibungen der Einzelkomponenten unter Berücksichtigung des VISION-UML-Profiles in typisierte UML-Komponenten überführt. Diese Komponenten wurden im nächsten Entwurfsschritt im UML-Werkzeug unter Berück-

sichtigung der Schnittstellenattribute zu einem Gesamtmodell des Subsystems kompositioniert. Die Modellierungsmethodik erlaubte hierbei sowohl eine frühe Untyped-Modellierung als auch eine Typed-Modellierung. Es wurde ein werkzeugunterstützter Ansatz entwickelt, welcher eine nahtlose Überführung von Untyped- in Typed-Modelle durch Nutzerinteraktion umsetzte, und dabei eine automatisierte Modellanpassung benutzerspezifischer Ressourcen unter Verwendung der im Projekt entwickelten Profilerweiterungen realisierte. Nachfolgend wurde dieses Subsystemmodell durch Exportfilter in das IP-XACT-Format als Systemmodell exportiert (*IP-XACT-Designfile*), um eine nahtlose Weiterverarbeitung mit Partnerwerkzeugen und im späteren Verlauf die nahtlose Integration in Wertschöpfungsketten zu erlauben. Die Wahrung der Semantik des ursprünglichen Modells wurde dabei durch die VISION-Profilerweiterungen ermöglicht.

Des Weiteren wurde die Exportfunktionalität dahingehend erweitert, dass ausführbare Modelle aus dem zugrundeliegenden Metamodell abgeleitet werden konnten. Dabei wurde von demselben Ecore-Modell ausgegangen, welches auch bei der Systemkomposition bzw. bei der Generierung der resultierenden IP-XACT-Beschreibung verwendet wurde. Im Verlauf des Projekts wurden von Projektpartnern bereits Werkzeuge zur automatischen Generierung von einzelnen SystemC-Modulen aus abstrakten Komponentencharakteristika, die in IP-XACT spezifiziert sind, entwickelt. In Kombination mit dieser Komponentengenerierung konnte nun anhand des durch die Systemkomposition in UML beschriebenen und nach IP-XACT transformierten Systemmodells die Verbindungen der Komponenten zu einem ausführbaren Gesamtsystemmodell erzeugt werden. Dabei konnten auf hoher Abstraktionsebene (*Untyped-Modellierung*) generische Verbindungsstrukturen oder CP-Verbindungen (*communicating processes*) als Kommunikationsarchitektur in SystemC generiert werden. Für ein verfeinertes Systemmodell bestand aber auch die Möglichkeit, Bus-akkurate Kommunikationsstrukturen inklusive komplexer Protokolle oder sogar zyklengenaue Kommunikationsmodelle (AMBA, CAN, FlexRay,...) zu instanzieren (Abbildung 8).



**Abbildung 8: Verfeinerung der Systemmodelle und Generierung eines Virtualen Prototyps**

Zur Generierung des ausführbaren Systemmodells wurde das ebenfalls Eclipse-basierte Framework *oAW* (*openArchitectureWare*) benutzt. Dieses Framework liefert eine Unterstützung beim Parsen beliebiger Modelle, stellt Sprachen bereit zur Überprüfung und Transformation von Modellen und generiert Code basierend auf diesen Modellen. Durch eine Template-basierten Modell-zu-Text-Transformation wurde aus dem UML/Ecore-Eingabemodell der Systemkomposition ein SystemC-Modell generiert.

Dieser Ansatz wurde am Beispiel des Projektdemonstrators angewendet und getestet. Die aus der Anwendung zur Verkehrszeichenerkennung resultierende Plattform ist in Abbildung



- 
- Eignung der Methodik zur Beschreibung komplexer verteilter Systeme. Dies beinhaltet sowohl die Möglichkeit zur Abbildung von Elementen solcher Systeme (Komponenten, Subsysteme, Kommunikationsstrukturen) als auch von Strukturen (Systemhierarchien).
  - Einheitlichkeit der Methodik, wobei möglichst auf Standards aufgebaut werden sollte. Dies erleichtert den Austausch von IP mit Kunden und Drittanbietern. Diese Anforderung wurde durch intensiven Austausch mit dem IP-XACT Standard des SPIRIT Konsortiums über das Projekt SPRINT und auch über Mitarbeit in dem Konsortium gewährleistet.
  - Exaktheit und Intuitivität der Beschreibungen. Dies wurde durch die Vermeidung von Redundanzen und ständigen Abgleich mit Infineon-Anforderungen erreicht.
  - Eignung der Beschreibungen zur automatischen Weiterverarbeitung. Dieser zentrale Punkt schließt insbesondere die Eignung für die automatische Erzeugung von Code mit ein. Die Eignung für die Codegenerierung und die Anpassung an Infineon-Anforderungen waren die beiden entscheidenden Gründe, warum bestehende Standards, insbesondere der IP-XACT Standard, nicht vollständig ausreichten. Die in VISION entwickelte Beschreibungs- und Generierungsmethodik kann auch interne Eigenschaften von IP abbilden und ist daher für die Codeerzeugung geeignet.

Im Rahmen des VISION Projekts wurde dabei zunächst ein Fokus auf statische Eigenschaften (also kein Verhalten, sondern Schnittstellen, Registersätze und Verbindungen) und die Abstraktionsebenen RTL und höher gelegt. Die Datenmodelle werden aber ständig erweitert, um auch Informationen von anderen Abstraktionsebenen aufnehmen zu können. Zur einfachen Anwendung der Methodik wurde eine grafische Benutzerschnittstelle für die Bearbeitung von Komponentenbeschreibungen entwickelt.

Die Beschreibungsmethodik enthält Datenmodelle für Komponenten, Systeme (Komponenteninstanzen und Verbindungen), Bussen und Interfacedefinitionen (separate Beschreibungen von Schnittstellen, die wiederverwendet werden können, z.B. die Signale eines Buses).

Zur Validierung und beispielhaften Anwendung wurden dann in einem weiteren Schritt reale Komponenten mit dieser Beschreibungsmethodik modelliert, wodurch die Methodik weiter verfeinert werden konnte. Durch den zusätzlichen ständigen Abgleich der Methodik mit Infineon Geschäftsbereichen (vgl. Beitrag 4.1.2) kann die Beschreibungsmethodik somit einige besonderen Eigenschaften aufweisen, von denen einige beispielhaft im Folgenden genannt werden sollen:

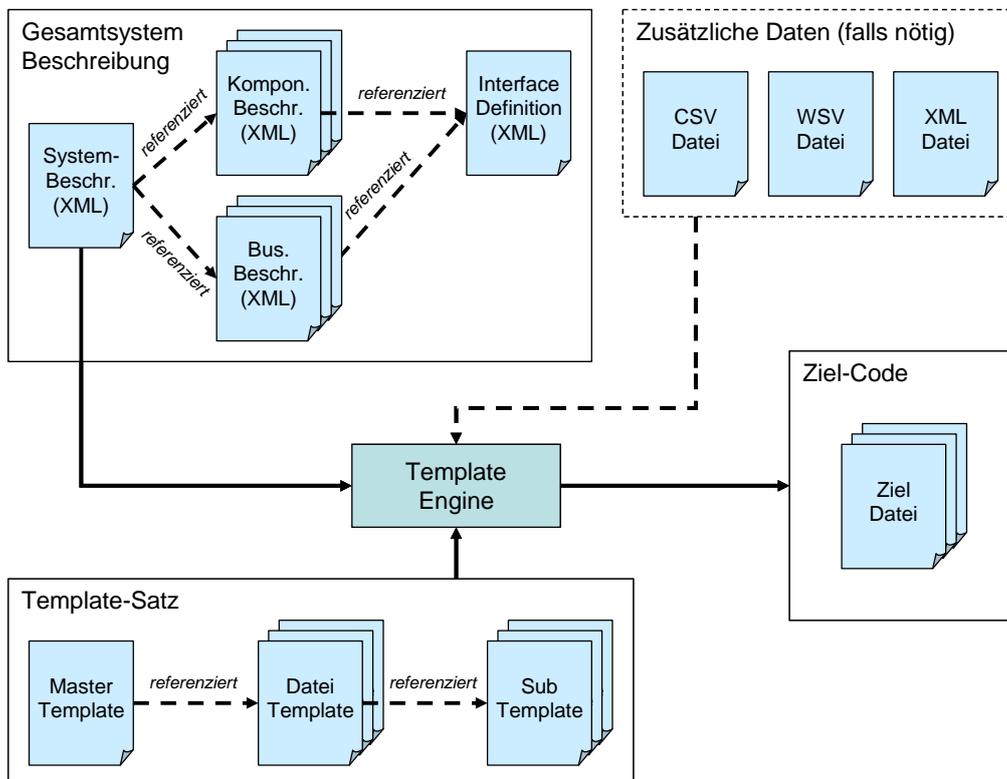
- Erweiterte Dokumentationsmöglichkeiten. Neben der Erzeugung von Code für die einzelnen Entwicklungsschritte ist ein wichtiges Generierungsziel auch die Dokumentation. Zu diesem Zweck erlaubt die Beschreibungsmethodik auch formatierten Text für die Dokumentation von Elementen einer Beschreibung. Diese Formatierung wird dann bei der Generierung übernommen, so dass ansprechende Dokumente erzeugt werden können.
- Parametrisierung. Da eine Beschreibung in der VISION-Methodik die Rolle einer formalen Spezifikation einnimmt, muss diese auch Produktvariationen abdecken können. Hierzu wurde ein leistungsfähiger Parametrisierungsmechanismus entwickelt und in die Methodik integriert, der (durch entsprechendes Setzen der Parameter) die Erzeugung von Code für mehrere Produktvarianten aus einer einzigen Beschreibung ermöglicht.
- Kapselung durch API. Der Zugriff auf die Daten der Beschreibungsmethodik geschieht prinzipiell über eine API, in der bereits grundlegende Konsistenzprüfungen stattfinden. Dadurch werden bereits bei der Eingabe grundlegende Fehler in der Spezifikation vermieden.

Basierend auf der verfeinerten Beschreibungsmethodik wurde im dritten Schritt des Beitrags die Generierungsmethodik entwickelt, die es erlaubt, aus Beschreibungen praktisch beliebigen Code zu erzeugen. Diese Codegenerierungsmethodik ist Template-basiert, was ein sehr

schnelles Erstellen von Generatoren ermöglicht. Hierbei können die Templates auch modular aufgebaut werden, was eine Wiederverwendung von Generatorteilen ermöglicht.

Eine weitere Besonderheit der Generierungsmethodik ist ihre Flexibilität, die auch die Einbindung von Informationen erlaubt, die nicht in der Beschreibungsmethodik spezifiziert wurden. Dies erweitert die Anwendungsmöglichkeiten der Generierung beträchtlich und erlaubt den Einsatz der Werkzeuge auch bei Prozessschritten, wo (noch) nicht alle nötigen Daten in der Beschreibungsmethodik abgebildet werden, weil diese z.B. sehr spezifisch für diesen Schritt sind. Damit ist es möglich, Beschreibungen in der VISION Methodik mit nicht in der Methodik enthaltenen Daten zu kombinieren, um Zielcode zu erzeugen.

Ein Überblick über den Generierungsflow, der die allgemeine Erzeugung von Zielcode aus der Beschreibung eines komplexen Systems bzw. Systemverbundes darstellt, ist in Abbildung 10 gezeigt.



**Abbildung 10: Generierungsflow basierend auf Beschreibungsmethodik**

Zusammenfassend entstand in diesem Beitrag eine leistungsfähige und flexible Beschreibungs- und Generierungsmethodik, die durch die Formalisierung von Eigenschaften die automatische Erzeugung von Zielcode erlaubt. Dadurch entstehen folgende Vorteile:

- Direkte Zeitersparnis durch automatische Erstellung von Code im Gegensatz zu manueller Eingabe. Je nach Komplexität des Codes kann hier mit einer Zeitreduktion um bis zu 99% pro Datei gerechnet werden (teilweise einige Sekunden gegenüber mehreren Stunden).
- Fehlervermeidung durch Konsistenz der formalen Informationen über mehrere Prozessschritte, da alle Zielformate aus der gleichen Beschreibung erzeugt werden können.

- Mittelbare Zeitersparnis und Fehlerreduktion durch automatische von Änderungen. Da die Codeerzeugung automatisch erfolgt, müssen Änderungen nicht von Hand bei allen Prozessschritten nachgezogen werden.

## Aufgabe 1.2: Ableitung der Kommunikationstopologie und Netzwerkarchitektur

### Beitrag 1.2.1: Abbildung von abstrakten Anforderungen des verteilten Systems auf Parameter des Verbindungsnetzwerks und Entwicklung entsprechender Simulationsmodelle

Im Beitrag 1.2.1 wurden von Bosch Methoden erforscht, die es erlauben, abstrakte Anforderungen und Randbedingungen des Systems in konkrete Parameter der zu realisierenden Verbindungsstruktur umzusetzen. Hierzu werden Ansätze zur Abbildung komplexer System-Anforderungen (etwa Sicherheits- oder Zuverlässigkeitseigenschaften) auf reale Parameter des Verbindungsnetzwerks untersucht und Simulationsmodelle abgeleitet, die auf abstrakter Ebene Aufschluss über das Kommunikationsverhalten des Systems geben und so die Grundlage für eine Bestimmung von Parametern zur Auswahl, Dimensionierung und Konfiguration einer optimierten Netzwerkarchitektur liefern.

Im Rahmen des Beitrags 1.2.1 wurden von Bosch Anforderungen bezüglich der Beschreibung komplexer Netzwerkstrukturen sowie deren Konfiguration und Parametrisierung identifiziert. Darauf aufbauend erfolgte die Spezifikation einer geeigneten Datenbasis zur Abbildung der Anforderungen in Form eines XML-Schemas. Aufsetzend auf dieser Datenbasis erfolgte die Entwicklung und Implementierung einer Werkzeugumgebung zur Beschreibung und Analyse komplexer Netzwerkstrukturen, welche auf abstrakter Ebene eine Untersuchung des Einflusses der Konfiguration und Parametrisierung des Verbindungsnetzwerks auf das Systemverhalten ermöglicht. Hierzu wurde ein Modellierungswerkzeug auf Basis von Java/Eclipse realisiert, welches eine Beschreibung von Systemkomponenten sowie eine Modellierung und Parametrierung von Netzwerkszenarien auf Basis einer interaktiven graphischen Benutzeroberfläche unterstützt. Die Umgebung stellt verschiedene Sichten auf das Systemmodell zur Verfügung und liefert damit die Grundlage für eine Analyse der Freiheitsgrade des Verbindungsnetzwerks auf abstrakter Modellebene. Die Umgebung liefert ferner die Grundlage für die Ableitung ausführbarer Simulationsmodelle, die für eine weitergehende Analyse funktionaler wie auch nicht-funktionaler Eigenschaften des Netzwerks (etwa Performanzverhalten) verwendet werden können. Abbildung 11 zeigt einen Screenshot der graphischen Benutzeroberfläche des Modellierungswerkzeugs. Das Werkzeug bietet neben einem graphischen Structureditor (A), einer Source-Code- und Hierarchie-Sicht (B) sowie einem Property-Editor (C) ferner eine flexible Abbildung von Software-Komponente auf Prozessor-Einheiten innerhalb der Hardware-Struktur (D).

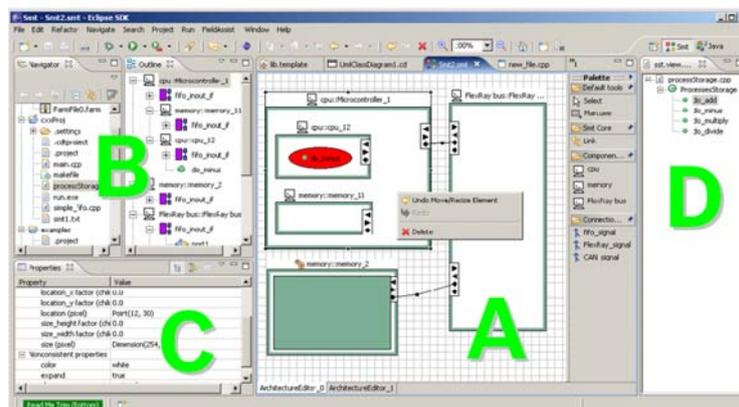


Abbildung 11: Modellierungswerkzeug für Netzwerkarchitekturen

Die Werkzeugumgebung wurde erfolgreich für eine Beschreibung und Untersuchung komplexer FlexRay-Cluster angewendet, welche das innerhalb der nachfolgenden Arbeiten betrachtete Anwendungsszenario lieferten.

Im weiteren Verlauf der Arbeiten wurde das Modellierungswerkzeug um wichtige Komponenten und Schnittstellen erweitert. Diese umfassen einen Bibliothekseditor zur Modellierung von FlexRay-Bibliothekskomponenten, welcher eine automatische Konsistenzprüfung der relevanten Parameter unterstützt, ein Werkzeug zum Import und Export von FIBEX-Beschreibungen, welches eine automatische Extraktion der relevanten Komponenten eines FlexRay-Clusters, der zugehörigen Message-Konfiguration und Timing-Parameter aus dem FIBEX Datenformat ermöglicht, sowie Schnittstellen, welche einen Import und Export von FlexRay-Konfigurationsdateien realisieren und dabei Mechanismen zur Validierung der Konsistenz der Konfigurationsdaten zur Verfügung stellen. Aufsetzend auf einem bestehenden FlexRay SystemC Referenzmodell ermöglicht das Werkzeug ferner die Generierung eines virtuellen SystemC Prototypen für das beschriebene FlexRay-Szenario.

Zusätzlich wurde das Modellierungswerkzeug um einen leistungsfähigen Analyseansatz ergänzt, welcher eine effektive Visualisierung und Bewertung von komplexen Kommunikationsabläufen, wie sie in komplexen Netzwerkszenarien typischerweise auftreten, überhaupt erst möglich macht. Für den Umgang mit komplexen Automotive-Simulationsszenarien wurde eine feingranulare Kopplung zwischen Matlab/Simulink und SystemC realisiert, die auf Basis einer Erweiterung des OSCI SystemC-Kernels eine effiziente und graphisch unterstützte Anbindung von Matlab/Simulink Stimuli-Generatoren und Testsequenzen an ein SystemC Simulationsmodell ermöglicht. Da Matlab/Simulink im Automobilbereich einen weit verbreiteten Ansatz zur Spezifikation komplexer Abläufe darstellt, eröffnet diese Kopplung die Möglichkeit für weitreichende Analysen der Kommunikation und Interaktion in Automobilnetzwerken. Abbildung 12 zeigt einen Überblick über die entstandene Werkzeugumgebung.

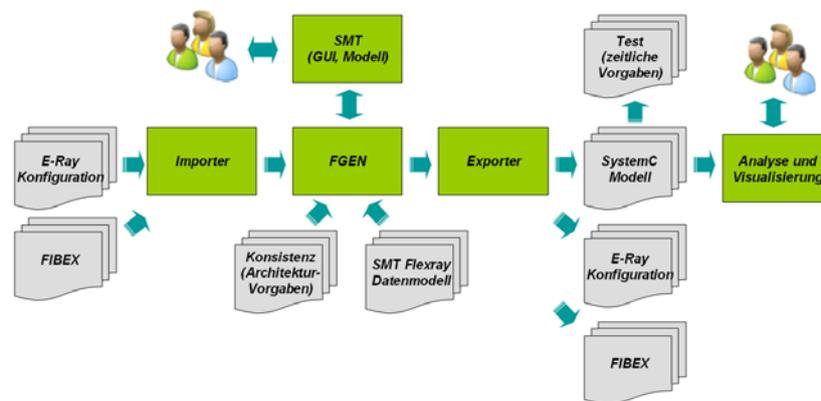


Abbildung 12: Tool-Flow

Aus diesen Komponenten resultiert als zentrales Ergebnis der Bosch Arbeiten in Aufgabenpaket A1.2 ein ganzheitliches Vorgehen zur Modellierung, Analyse und Exploration komplexer FlexRay-System Szenarien auf hoher Abstraktionsebene mit Möglichkeit für eine automatisierte Generierung eines virtuellen SystemC Prototypen. Abbildung 13 stellt die Methodik im Überblick dar.

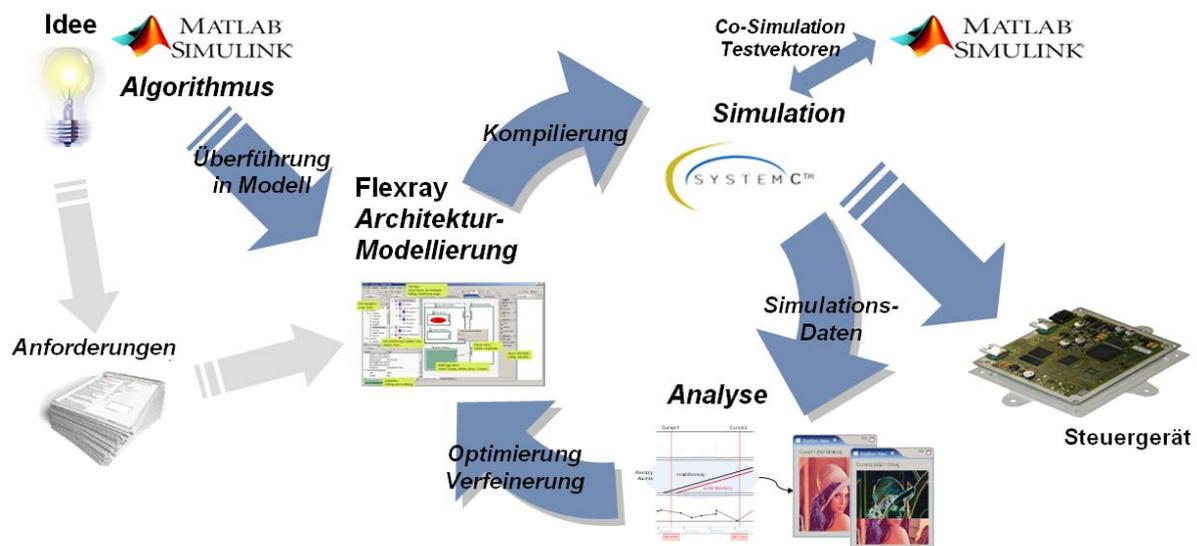
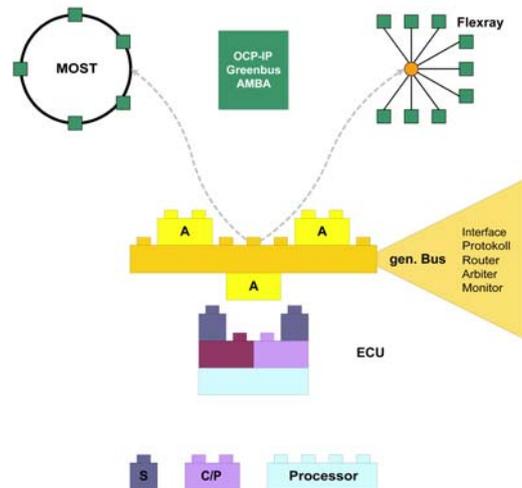


Abbildung 13: Systementwurfsmethodik für FlexRay-Szenarien

### Beitrag 1.2.2: Abbildung von Kommunikationskanälen und automatisierte Erstellung von Kommunikationsmonitoren zur Simulationsüberwachung von zeit- und ereignis-gesteuerten Zugriffsverfahren

Kommunikationsmechanismen spielen vor allem in verteilten Systemen eine entscheidende Rolle, um die unterschiedlichen Komponenten korrekt miteinander zu verbinden und deren Synchronisation sowie den Datenaustausch zu gewährleisten. Derzeit wird oftmals an Hand der Erfahrung des einzelnen Entwicklers entschieden, ob die Art der Kommunikation zeit- oder ereignisgesteuert ablaufen soll - ein Überwachungsschritt und eine automatisierte Beurteilungsinfrastruktur fehlen. Ziel dieser Arbeiten bei Cadence war es eine busähnliche Kommunikationstopologie aufbauen zu können, die es frühzeitig im Entwicklungsprozess ermöglicht, ein in SystemC implementiertes simulationsfähiges Modell aufzustellen. Weiterhin wurden im Rahmen dieses Arbeitspaketes Monitore entwickelt, die die Beobachtung und Überwachung der Kommunikation, d.h., des Zusammenspiels einzelner Komponenten innerhalb eines verteilten Systems übernehmen. Dies geschah für eine Reihe ausgewählter Busprotokolle. Die Kommunikation zwischen einzelnen Funktionskomponenten fand in diesen Arbeiten ausschließlich über diese Busse statt. Hierzu wurde eine Methodik für die Beschreibung der in SystemC zu implementierenden Busse erarbeitet.

Ziel dieser Funktionskomponenten war es eine Kommunikationstopologie aufbauen zu können, wie sie in einer Plattform oder gar in einem automobilen Netzwerk verwendet wird. Letzteres besteht neben diversen Bussen, die unter einander über so genannte *gateways* miteinander verbunden sind, und einzelnen Steuergeräten (*electronic control unit ECU*), deren Struktur sich ähnelt und aus Bausteinen, wie einem Businterface, einem Controller, Speicher und Sensoren bzw. Aktoren besteht. Hier soll es Ziel sein, neben einem generischen Bus ein generisches Steuergerät durch Konfiguration aus diesen Basiskomponenten erstellen zu können.



**Abbildung 14: Zusammenstellung und Aufbau einer Kommunikationstopologie aus generischen Komponenten**

Der generische Prozessor (GP, *processing unit*) entspricht einem Controller auf dem C-Programme ausgeführt werden können. Er wird wie alle anderen Komponenten auch in Form eines wohl definierten templates in SystemC beschrieben. Die einzelnen Controller sind durch generische Busse (GBus) miteinander verbunden.

Weitere Module zum Modellieren eines generischen Steuergerätes sind Sensoren und Aktoren, die ähnlich zu reinen Datenproduzenten oder –verbrauchern zunächst als einfach Speicher implementiert werden können, die zusätzliche algorithmische Funktionen zur Manipulation der Daten beinhalten.

Für eine Erweiterung des generischen Busses hinzu einer spezifischeren Beschreibung von Busprotokollen wurden weitere einzelne Komponenten zur Verfügung gestellt:

- Arbitr, für komplexere Arbitrierungsverfahren
- Router, für die Aufteilung und Zusammenführung von Datenpfaden
- Protokoll, Bus spezifisches Verhalten
- Monitor, zur Überwachung der Busaktivität und als Analyseinterface

Die einfache Kommunikation wird weiterhin auf dem generischen Busansatz ermöglicht.

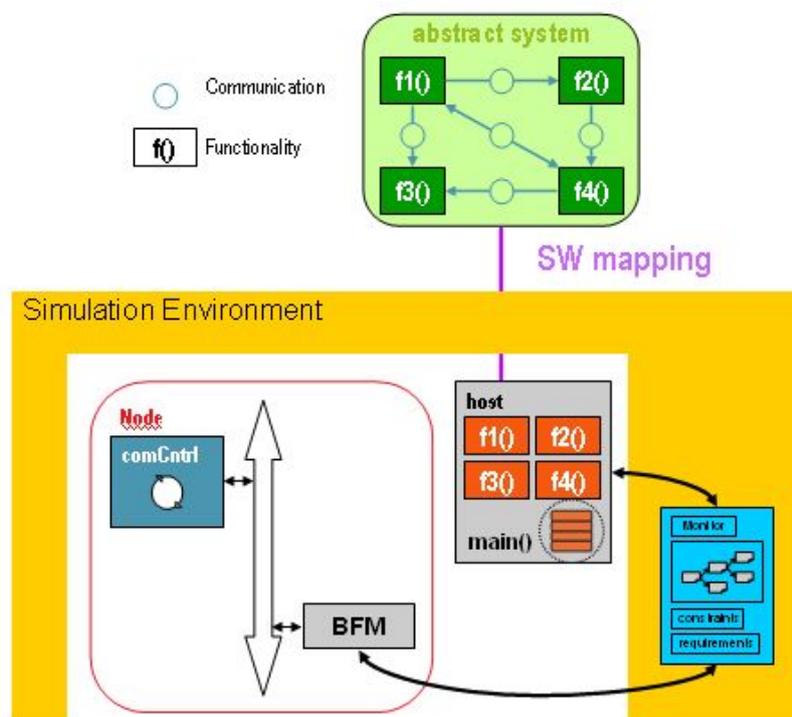
Hierzu wird zunächst aus den zuvor beschriebenen „Bausteinen“ die gewünschte Topologien zusammengestellt, wie es beispielhaft in Abbildung 14 dargestellt ist.

In diesem Beispiel wird aus dem generischen Prozessor, zwei Sensoren und einem Datenerzeuger (producer P) sowie einem Datenkonsumenten (consumer C) eine generische ECU zusammengebaut, die dann über einen nur abstraktionsspezifischen Adapter an den generischen Bus angeschlossen wird. Der Bus, kann wie hier dargestellt dabei verschiedenen Topologien, wie eine Ringstruktur (z.B. MOST) oder Netzstruktur (z.B. Flexray) beschreiben und erlaubt es die gewünschte Anzahl an Komponenten anzuschließen.

Die Simulationumgebung wird dann auch zur Stimulierung des generischen Modells und zur Erfassung des Kommunikationsflusses durch Kommunikationsmonitore nach den spezifischen Anforderungen aufgebaut. In der nächsten Phase werden wir den Ansatz der generischen Kommunikationskomponenten auf reale Bustopologien, in erster Linie den Flexray Bus, anwenden und dabei insbesondere auch die Konfigurierbarkeit der generischen Komponenten überprüfen. Das Ziel dieser Aufgabe ist es herauszufinden, ob eine automatische Generierung einer Kommunikationstopologie auf diese Art erreicht werden kann. Grundlage für die Parametereinstellung sollen dabei neben Busprotokoll spezifischen Anforderungen auch weitere Designbedingungen sein.

Zur Validierung und Verifikation wird eine Simulationsumgebung benötigt. Diese soll, wenn möglich in ihrer Grundstruktur unverändert bleiben, unabhängig von der Abstraktion des jeweiligen Modells. Hinzu kommt eine weitere Aufgabe, die von dieser Simulationsumgebung wahrgenommen werden kann: die funktionale Simulation zur Bewertung des Systemverhaltens, um Entscheidungen über die Effektivität der Kommunikation zwischen den einzelnen Komponenten sowie über die Architekturselektion unterstützen zu können.

In unserem Vorgehen können wir unter Verwendung der funktionalen Beschreibung des Systemverhaltens generiert aus dem SysML Model, über dessen Entstehung im Bericht zu Arbeitspaket AP1.1 referiert wurde, und unter Verwendung eines Generators eine Simulationsumgebung, wie sie in Abbildung 15 dargestellt ist, aufbauen.



**Abbildung 15: Simulationsumgebung für eine host code Ausführung des Systemverhaltens, hier an Hand des Beispiels für einen Bremsknoten**

Die Simulationsumgebung selbst besteht aus einem software UVC, universal verification component, die aus einem Sequenztreiber, einem Monitor und diversen Kontrolleinheiten besteht. In diesem speziellen Anwendungsfall – host code execution - werden die einzelnen Funktionen in einem host Model, das einen Prozessor ersetzt, integriert werden. Sie können durch den Sequenztreiber der Verifikationskomponente in beliebiger Reihenfolge und mit zufällig generierten oder festgelegten Parametern aufgerufen werden und über ein funktionales Busmodell in das Design eingespeist werden. Prinzipiell ist diese Vorgehensweise über verschiedene Modellabstraktionen hin anwendbar. In dieser Arbeit und mit dem Fokus von Vision verwenden wir eine SystemC Repräsentation des Modells.

Grundsätzlich sind in unserer Vorgehensweise mehrere Simulationsschritte vorgesehen, die unterschiedlichen Aufgaben zugeordnet sind. Eine erste Simulation wird auf Grundlage des aus SysML generierten C++ Modells durchgeführt, um das Verhalten zu validieren. Eine weitere Simulation findet in unserer in AP 1.1 aufgezeigten Entwicklungsumgebung durchgeführt. Diese Simulation hat zunächst die Aufgabe die Modellübernahme aus dem SysML-Werkzeug zu verifizieren. Da es sich um verschiedene Semantiken der jeweiligen Modellbeschreibungen (SysML, Cadence Repräsentation) handelt, ist dieser Schritt notwendig. Um

das Simulationsverhalten überprüfen zu können, werden in der Entwicklungsumgebung bereits eigene Monitore generiert, die die Sequenz der Funktionsaufrufe sowie die entsprechenden Parameterwerte überprüft. In einem weiteren Schritt werden diese Monitore auch in der SystemC Synthese mitgeneriert.

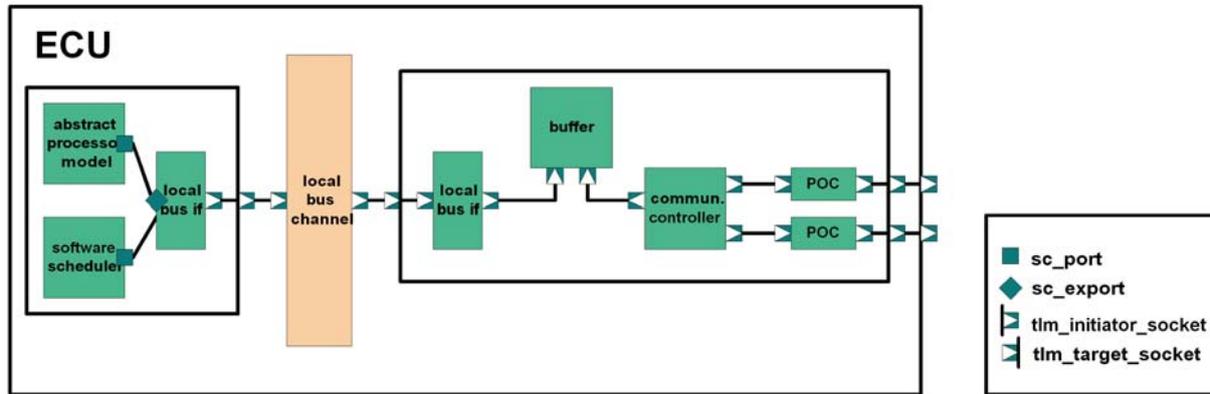


Abbildung 16: Vereinfachte Struktur eines abstrakten FlexRay Knotens

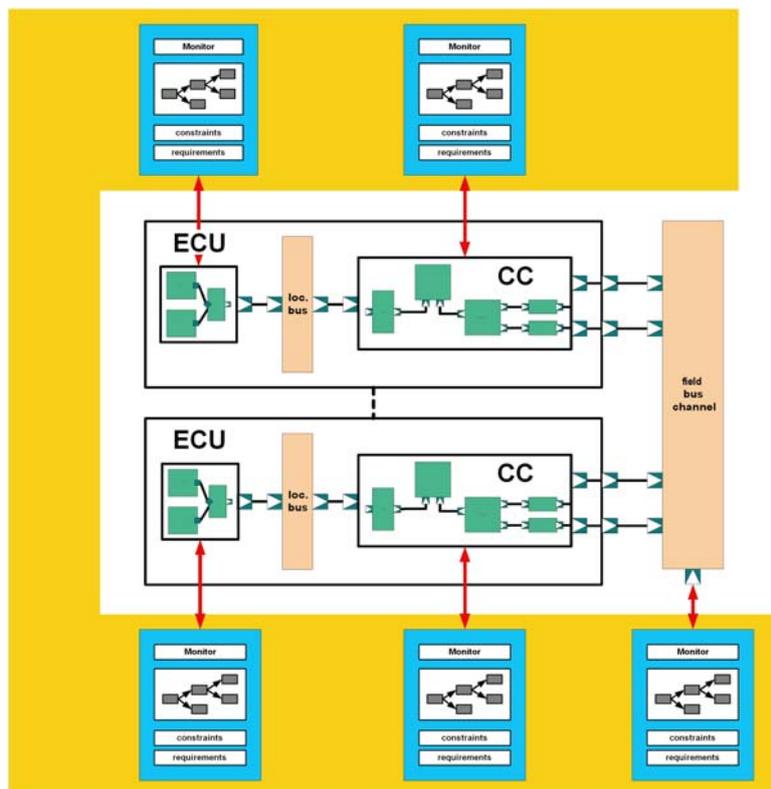


Abbildung 17: Simulationsumgebung für ein aus zwei Knoten bestehendes Kommunikationsmodell

In Abbildung 17 ist exemplarisch der Aufbau einer Simulationsumgebung für ein Kommunikationsmodell eines Systems bestehend aus zwei Knoten und einem abstrakten Feldbus, wie z.B. FlexRay, dargestellt. Die beiden dem host zugeordneten Simulationskomponenten

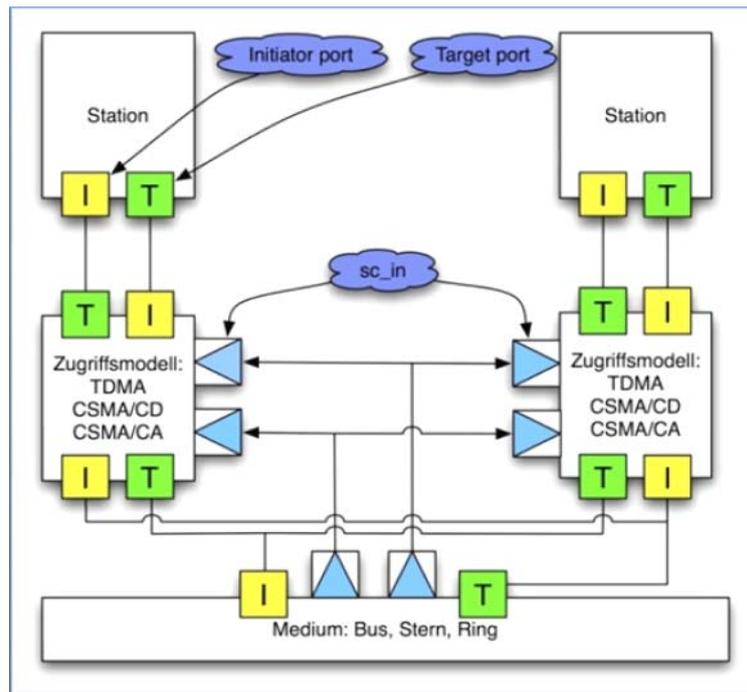
werden im master Modus betrieben und generieren bzw. konsumieren Daten, die dann über den Feldbus gesendet oder empfangen werden. Der Kommunikationskontroller CC steuert gemäß der Kommunikationsmatrix den Datenaustausch, sobald und so lange Daten vorhanden sind. Um diese Arbeit überwachen zu können werden sowohl auf den CC wie auch auf den Bus selbst Monitore aufgesetzt, die die Kommunikation überwachen und sowohl die Designanforderungen als auch mögliche Einschränkungen überwachen. Hierzu sind auch im Design „Minimonitore“ eingebaut die Simulatoren der Simulationsumgebung mit den relevanten Daten versorgen.

### **Beitrag 1.2.3: Automatische Abbildung von Kommunikationskanälen auf Busstrukturen unter Berücksichtigung zeit- und ereignisgesteuerter Zugriffsverfahren**

Im Rahmen des diesem Beitrag zugehörigen Aufgabenpakets wurden vom FZI Methoden und Werkzeuge entwickelt, die eine Template-basierte Verfeinerung von Kommunikationen auf Busstrukturen, sowie die Generierung eines dem verfeinerten Modell entsprechenden virtuellen Prototyps zur simulativen Bewertung der unterschiedlichen Zugriffsverfahren ermöglichen.

Bei den Kommunikationen handelt es sich zunächst um Punkt-zu-Punkt-Verbindungen, welche durch die entwickelte Verfeinerungsstrategie auf konkrete Zugriffsverfahren abgebildet werden. Zunächst werden die Modelle durch sogenannte Kommunikations-Templates repräsentiert, durch welche die Kommunikationsmechanismen charakterisiert werden, um sie für einen automatisierten Verfeinerungsprozess nutzen zu können. Durch die Kommunikations-Templates ist auch eine Parametrisierung und somit Konkretisierung der implementierten Modelle möglich. Für die Repräsentation der Kommunikations-Templates wird das standardisierte IP-XACT Format verwendet, da es zur Beschreibung von Kommunikationen gut geeignet ist, bzw. spezifische Erweiterungen an erforderlichen Stellen ermöglicht. Hierzu wurden charakteristische Merkmale, vorhandene Schnittstellen sowie notwendige Parameter der Kommunikationsmechanismen herausgearbeitet und mit Hilfe von IP-XACT beschrieben.

Darüber hinaus wurde ein abstraktes, zeit- bzw. ereignisgesteuertes Kommunikationsverfahren entwickelt, welches eingesetzt wird, bevor ein konkretes Kommunikationsprotokoll ausgewählt ist. Dies ist sowohl für die schrittweise Verfeinerung der Kommunikation, als auch für eine frühzeitige Evaluierung und Bewertung des Kommunikations- bzw. Gesamtsystemverhaltens hilfreich. Der Vorteil dieser Zwischenstufe ist, dass die Kommunikation bereits hinsichtlich ihres funktionalen und teilweise auch zeitlichen Verhaltens analysiert und bewertet werden kann, was mit abstrakten Punkt-zu-Punkt Verbindungen in diesem Umfang nicht möglich ist, ohne jedoch bereits eine feste Entscheidung hinsichtlich eines konkreten Kommunikationsprotokolls treffen zu müssen. Aus diesem Grunde wurden die entsprechenden Mechanismen als SystemC-Modelle implementiert und hinsichtlich einer Charakterisierung durch IP-XACT nutzbar gemacht. Die implementierten Modelle berücksichtigen sowohl ein zeitgesteuertes Kommunikationsverhalten, als auch einen ereignisgesteuerten und somit prioritätsbasierten Kommunikationszugriff.

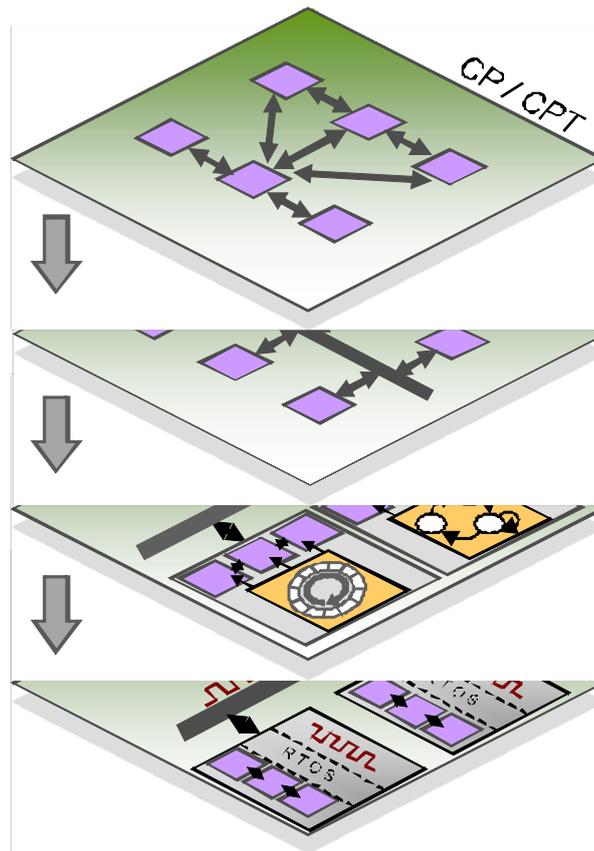


**Abbildung 18: Modell des abstrakten Kommunikationsverfahrens**

Um eine konfliktfreie Abbildung von Kommunikationen auf Kommunikationsressourcen zu gewährleisten, wurde außerdem ein Verfahren zur automatischen Allokierung und Bindung von Kommunikationskanälen entwickelt. Es verwendet die im Rahmen von Beitrag 2.1.2 entwickelte Analysemethodik, die das globale zeitliche Verhalten des Systems unter Berücksichtigung des internen Kontrollflusses der Prozesse und des Synchronisationsverhaltens der verwendeten Kommunikationsprimitive bestimmt, um den globalen zeitlichen Ablauf von Kommunikationen zu bestimmen und mögliche Überlappungen von Kommunikationsintervallen zu identifizieren. Identifizierte Überlappungen können bei Abbildung auf dieselben Kommunikationsressourcen zu Konflikten führen. Die Auswirkung der Konflikte auf das globale Echtzeitverhalten des Systems kann zur Verletzung von Zeitanforderungen führen. Zur Bereinigung dieser Konflikte wurde das Allokierungs- und Bindungsverfahren entwickelt, um eine Konfliktvermeidung statisch zu garantieren.

Weiterhin wurde eine Methodik zur semi-automatischen Verfeinerung von Kommunikationskanälen konzipiert, die aufbauend auf einem XML-Beschreibungsformat und API-Charakterisierungen das Kommunikationsprotokoll basierend auf Nutzerannotationen automatisiert im Analysemodell verfeinert. Hierbei wird, ausgehend von Kommunikationskanälen auf Applikationsebene, eine schichtenweise Transformation der Kommunikation durchgeführt und die Inklusion von protokolltypischen Synchronisationsprimitiven und Protokolldaten berücksichtigt.

Die entworfene Strategie zielt darauf ab, den Verfeinerungsprozess so weit wie möglich zu automatisieren. Die Verfeinerung basiert auf vorgegebenen Schablonen, was eine exakte Definition der Schnittstellen auf allen Abstraktionsebenen erfordert. Dies beinhaltet auch eine klare Festlegung bei der Umsetzung der Punkt-zu-Punkt-Kommunikation. Zusätzlich wurden die erforderlichen durchzuführenden Maßnahmen für jeden Verfeinerungsschritt erarbeitet und anhand von Beispielen validiert. Ein wichtiger Punkt ist hierbei die Komposition der Kommunikationsarchitektur, da kommunikationsspezifische Konfigurationen zwar durch die Konfiguration der entsprechenden (XML-) Schablone durchgeführt werden können (z.B. die Auswahl des gewünschten Zugriffsverfahrens), aber andere benötigte Informationen (z.B. die Vergabe von Kommunikations-IDs) nicht durch die Schablone erfasst sind. Hierfür muss ein zusätzliches applikationsspezifisches Konfigurationsschema zur Verfügung gestellt werden.



**Abbildung 19: Verfeinerungsstufen der Kommunikationsverfeinerung**

Im weiteren Verlauf wurde die automatische Generierung virtueller Prototypen aus den verfeinerten Modellen umgesetzt. Hierzu werden die innerhalb des Templates hinterlegten Informationen sowie das applikationsspezifische Konfigurationsschema verarbeitet und entsprechende SystemC-Dateien generiert, die dann zusammen mit IP-Modellen der entsprechenden Busse ein ausführbares SystemC-Modell ergeben. Die Implementierung erfolgte unter Eclipse. Um auch die direkte Verfeinerung eines virtuellen Prototyps zu ermöglichen, wurde das Verfahren erweitert, indem Informationen des virtuellen Prototyps in das Templateformat überführt werden. Dort kann dann die Verfeinerung durchgeführt und anschließend der verfeinerte virtuelle Prototyp generiert werden. Hierzu wurde eine SystemC Strukturparser entwickelt, der eine detaillierte Analyse des Designs, auch über Hierarchiegrenzen hinweg, durchführt und als Ergebnis eine Strukturbeschreibung ausgibt. Die Implementierung erfolgte ebenfalls unter Eclipse.

Als letzter Arbeitsschritt wurde ein Ansatz zur automatisierten Ableitung einer Netzwerk-Topologie unter Verwendung eines existierenden und bereits verfeinerten Systemmodells konzipiert. Für die Performanz und Zuverlässigkeit eines verteilten Systems spielt vor allem die Kommunikation zwischen den teilnehmenden Einzelkomponenten eine herausragende Rolle. Daher ist es wichtig, spezifisch für die auf dem verteilten System auszuführenden Anwendungen eine optimierte Topologie zu erhalten. Voraussetzung hierfür ist, dass das zeitliche Verhalten der Anwendungen auf den einzelnen Komponenten (ausgeführte Prozesse) im Gesamtzusammenhang erfasst werden kann.

Grundsätzlich existieren für die Analyse des Kommunikationsverhaltens von verteilten Systemen zwei Vorgehensweisen. Zum einen kann die Kommunikation durch formal-analytische Methoden erfasst werden, was allerdings schnell zu einer hohen Komplexität führen kann. Zum anderen können durch Simulation des Systems erforderliche Kenngrößen ermittelt werden. Dies ist aber in der Regel ein zeitintensiver Prozess und es muss beachtet werden, ob bestimmte Werte eventuell abhängig von dynamischen Eingabedaten und somit vom jeweili-

gen Simulationsdurchlauf sind. Deswegen wurde ein hybrider Ansatz entwickelt, der die Vorteile beider Analyseverfahren kombiniert, indem sowohl Informationen von formalen Analysemethoden, als auch solche, die aus Simulationen gewonnen werden, verarbeitet werden, um eine auf die jeweilige Anwendung optimierte Verbindungstopologie zu erhalten.

Auf der Basis eines SystemC-Modells zur Fahrzeugumfeldererkennung im Automobil wurde ein Konzept für den oben beschriebenen Ansatz erarbeitet. Darin wird zunächst das erwähnte SystemC-Modell in die zuvor gewählte ISS-Simulationsumgebung integriert. Anschließend wird durch ein kombiniertes Verfahren das zeitliche Verhalten der Automobilanwendung analysiert. Zum einen wird in einem formal-analytischen Ansatz, der auf der Analyse von ermittelten Kommunikationszeitpunkten basiert und in Arbeitspaket 2.1 entwickelt wurde, das Kommunikationsverhalten des Gesamtsystems ermittelt. Zum anderen wird aber auch ein auf Simulation basierender Ansatz verwendet, um die benötigten Laufzeiten der einzelnen Anwendungen auf der Zielarchitektur zu erhalten. Die Ergebnisse dieses hybriden Analyseansatzes werden dann in einem inkrementellen Explorationsprozess zur Bestimmung einer optimierten Topologie, spezifisch für die jeweiligen Applikationen des verteilten eingebetteten Systems, benutzt.

#### **Beitrag 1.2.4: Automatisierte Parametrisierung von Kommunikations-Templates zur optimierten Topologiebestimmung von vernetzten mikroelektronischen Systemen.**

In diesem Beitrag wurde von Infineon eine Methodik zur Spezifizierung von Kommunikationsschablonen entwickelt, die sowohl eine automatisierte Parametrisierung und damit eine Anpassung an industrierelevante Kommunikationsmechanismen und -protokolle aber auch eine Generierung von ausführbarem Code zur Schnittstellenadaption ermöglicht. Durch Simulation und Analyse dieser generierten Kommunikationssysteme bzw. Kommunikations-Templates konnte dann eine Exploration bezüglich der Komponentenparameter und Topologie durchgeführt werden.

Hierzu konzentrierten sich die durchgeführten Arbeiten zunächst auf die Identifizierung relevanter Parameter zur Erstellung verschiedener Kommunikationsstrukturen und deren Zusammenführung in Kommunikationsmechanismen, die unabhängig von der Verwendung einer bestimmten Protokollklasse sind.

Hierfür wurde das schon bestehende und besonders im SoC-Entwurf eingesetzte Schnittstellen-Spezifikationsformat IP-XACT hinsichtlich der Eignung zur Spezifikation dieser kommunikationsrelevanten Parameter untersucht. Hintergrund dieser Aktivitäten war das Ziel, kommunikationsrelevante Informationen und Charakteristika in die Schnittstellen der Kommunikationspartner zu transportieren und darzustellen, um somit eine Analyse der Kommunikation auf dieser Ebene zu ermöglichen. Mithilfe dieser Analysemethoden, die in Zusammenarbeit mit Arbeitspaket 2 entwickelt wurden, war es möglich, eine optimierte Kommunikationstopologie in verteilten Systemen zu bestimmen. In Zusammenarbeit mit dem zuständigen Konsortium wurde IP-XACT um die zur Parametrisierung von Kommunikations-Templates notwendigen Eigenschaften erweitert.

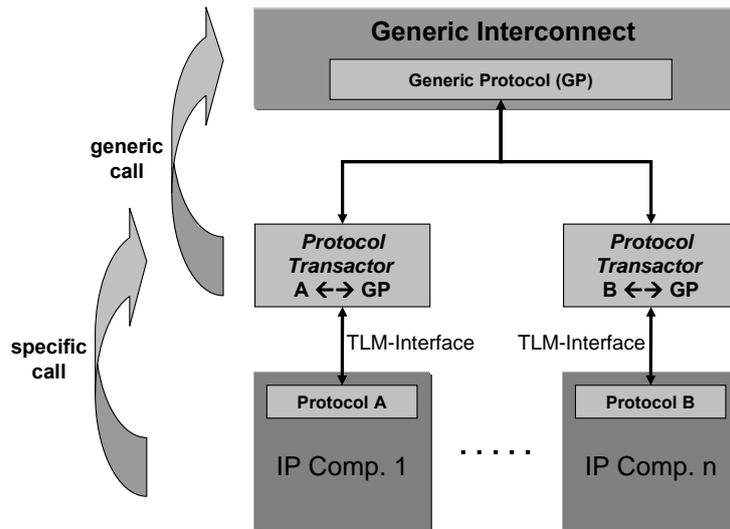


Abbildung 20: Architektur einer Schnittstellenanpassung

Zur Abstrahierung von Kommunikationsmerkmalen und damit auch der zugrunde liegenden Kommunikationstopologien wurde im VISION-Projekt der GreenBus-Ansatz der GreenSocs-Initiative, an dem die VISION-Partner auch maßgeblich beteiligt waren, als Ausgangspunkt verwendet und hinsichtlich der projektspezifischen Anforderungen verallgemeinert (siehe Abbildung 20). Diese Abstrahierung ermöglichte die Modellierung und Simulation kommunikationsrelevanter Eigenschaften auf unterschiedlichen Abstraktionsebenen.

Weitere Arbeiten konzentrierten sich auf die Erweiterung der Methodik zur Modellierung kommunikationsrelevanter Eigenschaften von einer Datenfluss-orientierten Sichtweise hin zu einer für eine Analyse bzw. Simulation der Kommunikation notwendigen Einbeziehung zeitlicher Charakteristika und Sequenzen (siehe Abbildung 21). Dabei wurden generische Kommunikationseigenschaften wie der grundsätzliche Aufbau von Protokollen in verteilten Systemen identifiziert, die in einem spezifikationsgetriebenen Anpassungsprozess während der Integration von Komponenten und Kommunikations-Templates durch die jeweils spezifischen Merkmale ergänzt und implementiert werden.

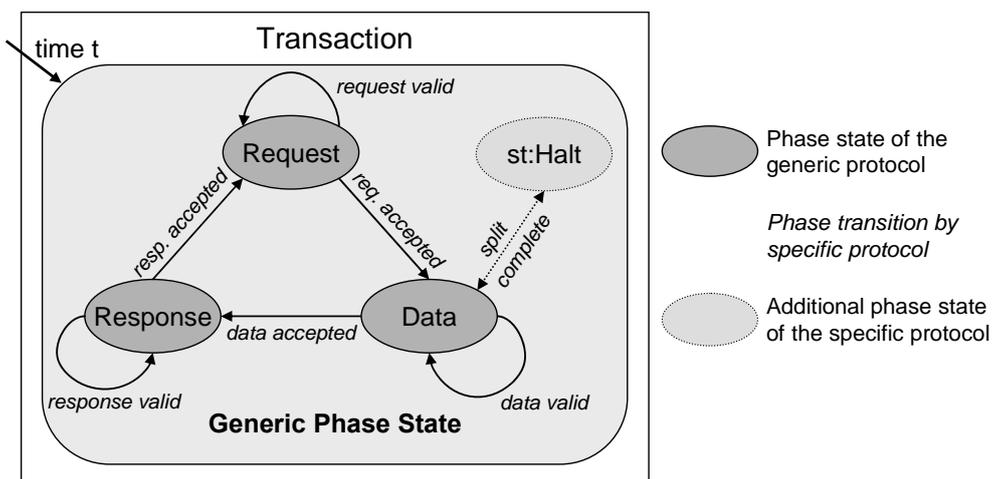


Abbildung 21: Kontrollfluss-orientierte Spezifikation generischer Kommunikationsprotokolle

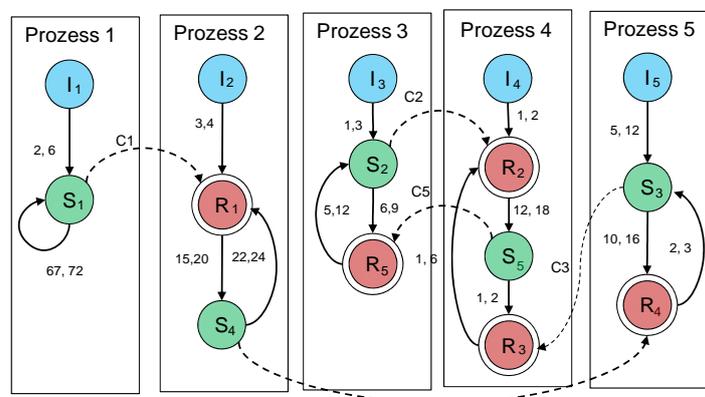
Zur verbesserten Interoperabilität und zur Unterstützung eines durchgängigen Entwurfsablaufs wurde diese Spezifizierung wie auch der generische Aufbau einer Transaktion in einer IP-XACT-Schnittstellenbeschreibung gekapselt.

Im weiteren Verlauf sollte diese Beschreibungsmethodik durch die automatisierte Generierung eines verteilten Systems inklusive der relevanten Kommunikationsmechanismen, wie

z.B. des zu verwendenden Busprotokolls, validiert und hinsichtlich der Verwendung in industrierelevanten Anwendungen evaluiert werden. In diesem Zusammenhang wurde ein Modell des im System-on-Chip-Bereich oftmals eingesetzten AMBA-Protokolls auf hohem Abstraktionsgrad sowohl bezüglich der zu übertragenden Daten wie auch der zeitlichen Abfolge spezifiziert. Durch Verknüpfung der protokollspezifischen Daten mit generischen Datencontainern und des Protokollablaufs mit Kommunikationsprimitiven einer generischen Busstruktur konnte das Verhalten eines Busprotokolls modelliert und in Form eines virtuellen Prototypen in SystemC generiert werden. Ein Beispiel für eine solche Verknüpfung ist die Einordnung einer Busanfrage in den Bus-Scheduler gemäß den geltenden Regeln der Schedulingstrategie. Zusätzlich zu dem virtuellen Prototypen wurde eine komponentenspezifische API für den Zugriff auf diese Busstruktur generiert, die die Anpassung der protokollspezifischen Daten und Kontrollsequenzen an die generischen Daten und Sequenzen automatisiert.

Im weiteren Verlauf wurden die bestehenden Ansätze zur Kommunikationsanalyse verteilter Systeme, die im Arbeitspaket 2.1 erarbeitet wurden, erweitert, sodass der im Vorfeld dieser Arbeiten definierte Ansatz zur Spezifikation von abstrakten Protokollmechanismen in IP-XACT integriert werden konnte.

Dazu wurden eine gemeinsame Vorgehensweise sowohl zur applikationsspezifischen Kommunikationsanalyse mithilfe eines von einem CDFG-Graphen abstrahierten Systemmodells (Erkennung und Synchronisierung von Kommunikationszeitpunkten, siehe Abbildung 22) als auch zur Methodik der Spezifizierung von generischen Datentypen und Protokollphasen entwickelt. Die Kombination dieser Ansätze ist notwendig, da die Spezifikation des zeitlichen Verhaltens im IP-XACT-Format nur auf die einzelne Komponente fokussiert ist und somit keinen Bezug zum Gesamtsystem besitzt. Eine automatisierte Parametrisierung der verwendeten Kommunikationsschablone kann aber nur vorgenommen werden, wenn auch der Kontext zum Gesamtsystem existiert.



**Abbildung 22: Beispiel Kommunikations-Abhängigkeits-Graph aus Arbeitspaket 2**

Die für eine abstrakte Komponente bzw. Kommunikationsschablone spezifizierten Sequenzen von Protokollphasen und die in diesen Phasen jeweils zu übertragenden Daten stellen dabei jeweils Kommunikationszeitpunkte im Gesamtkontext der restlichen Komponenten und Bus-Instanzen dar (z.B. Bus-Request einer Komponente zum gewünschten Sendezeitpunkt, Bus-Grant durch den zentralen Arbitrer, usw.). Durch diese globale Sichtweise kann eine Konfliktanalyse unter Berücksichtigung der Protokollsequenzen durchgeführt werden, um gleichzeitige Zugriffe auf geteilte Kommunikationsressourcen zu erkennen und die dadurch notwendigen Wartezeiten der beteiligten Komponenten und Kommunikationsschablonen in die Analyse mit einzubeziehen (siehe Abbildung 23). Weiterhin kann dadurch in Abhängigkeit des Gesamtsystems eine geeignete Parametrisierung der abstrakten Kommunikationsschablone gefunden werden, die eine konfliktfreie Kommunikation erlaubt, z.B. höchstmögliche Senderate oder eine bestimmte benötigte Busfrequenz.

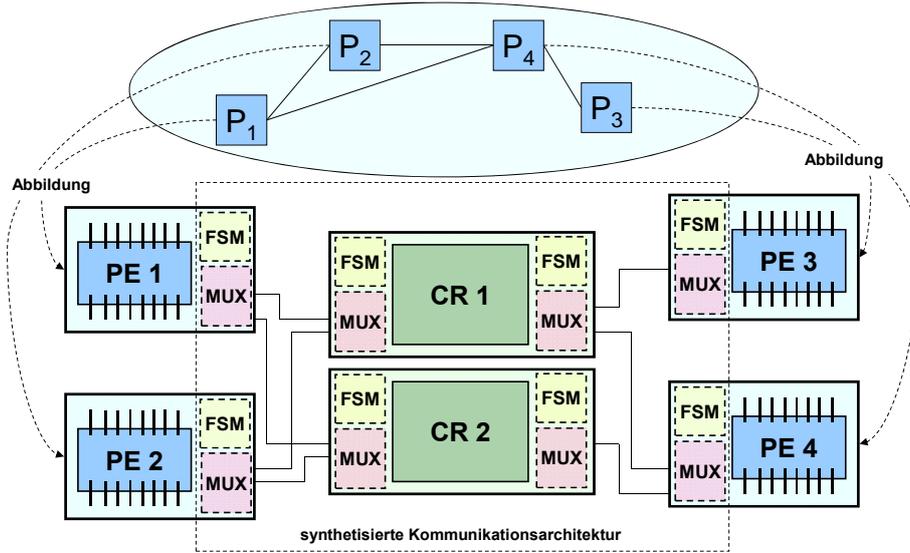


Abbildung 23: Konfliktfreie Abbildung von Kommunikationen auf die Architektur

## 4.2 AP2: Analyse und Bewertung der Eigenschaften verteilter Systeme

Im iterativen VISION Entwurfsablauf positionierte sich Arbeitspaket 2 als Bewertungsschritt, um eine frühzeitige Überprüfung von Entwurfsschritten im Hinblick auf Performanz, Qualität, Echtzeitfähigkeit und Konsistenz bereitzustellen. Dies ermöglicht bereits während der Spezifikation, verletzte Randbedingungen aufzuzeigen und automatisierte Verfeinerungsschritte strategisch zu steuern. Das Ziel des Arbeitspakets bestand deshalb darin, den Entwickler methodisch und durch geeignete Analyseverfahren darin zu unterstützen, das richtige System zu entwerfen. Hierfür wurden Verfahren erforscht, die es erlauben, eine systemübergreifende Analyse der Systemanforderungen durchzuführen. Dies beinhaltet Verfahren zur Analyse von Performanz und Kommunikationsverhalten des Gesamtsystems. Die Analyse kann sowohl dynamisch als auch statisch unter Einsatz analytischer Systemmodelle sowie unter Anwendung von (bzw. in Kombination mit) simulationsbasierten Techniken erfolgen.

Durch dieses Arbeitspaket werden erstmals Verfahren zur Verfügung gestellt, um bereits vor der Erstellung eines realen Prototypen eine Analyse über Subsystemgrenzen durchzuführen. Die Auswirkungen von Entwurfsentscheidungen, wie etwa die Auswahl einer Kommunikationstopologie, auf das Verhalten des Gesamtsystems können damit noch vor der Integration der Subsysteme in einen realen Prototypen untersucht werden. Neben der Modellierung der Umgebungsbedingungen wurden auch automatisierte Verfahren zur Kommunikationsanalyse verteilter, vernetzter Systeme entwickelt, die ein frühzeitiges Erkennen und Vermeiden von Engpässen in der Verbindungsstruktur ermöglichen. Systemanforderungen können nun bereits frühzeitig im Entwurfsprozess überprüft werden, da die Analyseverfahren auf einem abstrakten Systemmodell arbeiten. Eine schnelle und frühzeitige Untersuchung von Entwurfsalternativen noch vor der Erstellung eines realen Prototyps führt zu einer optimierten Systemarchitektur und Netzwerktopologie und vermeidet zeit- und kostenaufwändige Re-Designs.

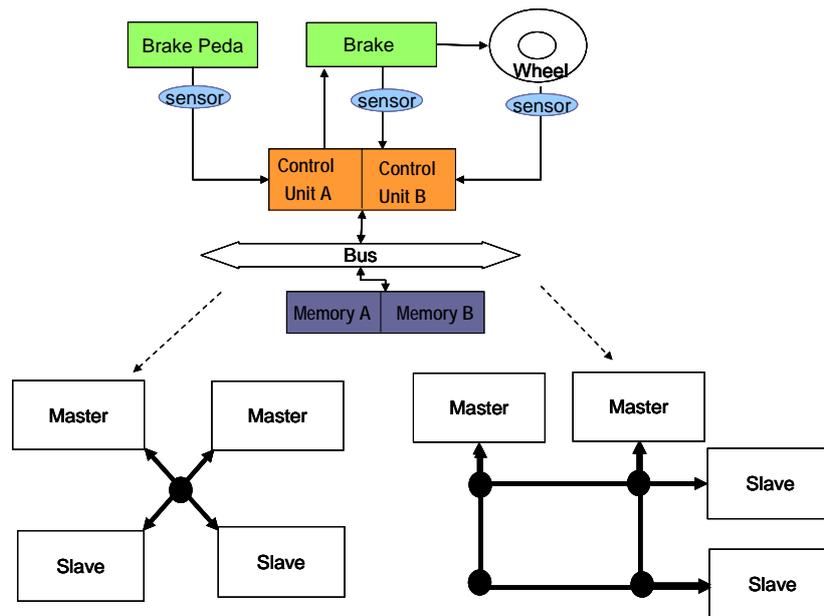
### Aufgabe 2.1: Performanz- und Kommunikationsanalyse

#### Beitrag 2.1.1: Analyse der Systemperformanz unter Berücksichtigung von unterschiedlichen Abstraktionsstufen der Kommunikation

Eine der wesentlichen Aufgaben des Systementwurfs besteht in der Analyse der erreichbaren Performanz und Kommunikationseffizienz. Erst die Analysierbarkeit auf Systemebene erlaubt eine detaillierte Spezifikation von Parametern wie etwa der Busbreiten, Cache-Größen, Taktfrequenzen und Ähnlichem unter Betrachtung der Auswirkungen auf das Gesamtverhalten. Daher gilt es, bereits in den frühen Phasen des Entwurfs, in der Regel auf einer hohen Abstraktionsstufe, entsprechende Untersuchungen durchführen zu können.

Da das Kommunikationsverhalten einer Anwendung ein wesentlicher Einflußfaktor auf die zu erreichende Systemperformanz ist, hat Cadence verschiedene Kommunikationsmodelle auf unterschiedlichen Abstraktionsstufen aufgebaut, welche die jeweiligen Möglichkeiten und Grenzen aufzeigen und die Extraktion von Modellierungsparametern erlauben.

Konkret wurden zwei Kommunikationstopologien in SystemC modelliert, eine „Hub and Spoke“ und eine „Crossbar“ Struktur. Im Weiteren wurde die Analyse der Kommunikationsstrukturparameter und der Vergleich der Performanz dieser beiden Strukturen durchgeführt und auf reale Bustopologien wie z.B. Flexray ausgedehnt. Hierbei wurde wie zuvor exemplarisch das Brake-By-Wire Beispiel benutzt.

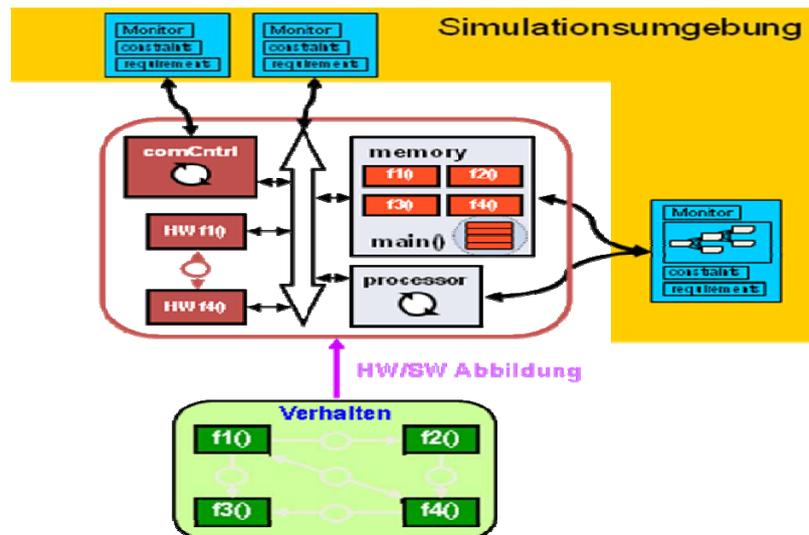


**Abbildung 24 – Abbildung des BBW Anwendungsbeispiels auf verschiedene Kommunikations-Topologien**

Die eingesetzten Analyseverfahren und Methoden wurden weiterhin auf ihre Wiederverwendbarkeit auf unterschiedlichen Abstraktionsebenen beurteilt. Aufbauend auf bereits bestehenden Performanz-Analysemethoden wurde eine Vorgehensweise untersucht und vorgeschlagen, die den Verwendungszweck der jeweiligen Implementierung von Kommunikation und Funktionalität auf unterschiedlichen Abstraktionsstufen vorschlägt.

Unter Verwendung einer übergreifenden Simulationsumgebung bestehend aus Datengeneratoren (Master) und –konsumenten (Slave) sowie von Überwachungseinheiten (Monitore). Bei dem vorgestellten Ansatz wurde die Funktionalität der Design-Anwendung sowie deren Struktur, die man auch als logische Architektur bezeichnen kann, in SysML modelliert und durch die Simulation des aus diesem Modell generierten C++ Code getestet.

Dieses Model kann somit als eine ausführbare Spezifikation angesehen werden. Die generierten C++ Funktionen, die die Design- Funktionalität realisieren, wurden eingebettet in die Simulationsumgebung verwendet, um eine Performanzanalyse und eine Architekturanalyse durchzuführen. Dabei überwachen einzelne Komponenten der Simulationsumgebung die Einhaltung der vorgegebenen Systemeigenschaften und Systemanforderungen unabhängig von ihrer Beschreibung – z.B. können Hardware Komponenten sowohl in SystemC oder jeder anderen RTL Sprache beschrieben sein. Dieser Ansatz erlaubt es zudem sowohl die Verfeinerungsschritte im iterativen HW/SW Partitionierungsprozeß als auch die Architekturanalyse der einzelnen Komponente zu unterstützen. Als ein Teil dieses Prozesses können unterschiedliche Modelle von Prozessoren integriert werden und die Kommunikation zwischen den einzelnen Komponenten im System gesteuert werden. Es wurden die relevanten Systemeigenschaften und –anforderungen analysiert, wie z.B. die periodische Auslieferung von Bremswertdaten über den Systembus unter den unterschiedlichen Auslastungsbedingungen des Datenverkehrs.



**Abbildung 25:** Durch eine partielle Realisierung der Funktionen in HW Modulen (in SystemC mit annotiertem Zeitverhalten) und der Ausführung der verbleibenden Funktionen auf dem Prozessormodell können unterschiedliche Varianten des HW/SW mapping simulativ bewertet werden.

Die Struktur der Simulationsumgebung sowie der hardware-software Kommunikationsmechanismus sind hierbei unabhängig vom benutzten Prozessormodell und der Sprache der Implementierung aufgesetzt. Während des Entwurfsprozesses verändert sich das Modell, so dass Varianten oder auch Modelle unterschiedlicher Abstraktion in einem komplexen System nebeneinander verwendet werden können wie z.B. ein zyklen-genaues Model oder RTL.

Der vorgestellte Prozess bietet eine durchgehende simulative Vorgehensweise, wie man von einer in SysML/UML beschriebenen funktionalen Beschreibung über verschiedene Verfeinerungsschritte zu einem Architekturmodell findet. Zur Bewertung und Überprüfung der Einhaltung des Anforderungsprofils wird jedes Modell durch Simulation in der gleichen Umgebung und unter Ausnutzung derselben Testsuite (corner test und zufällig generierten Stresstests) überprüft. Die Simulationsergebnisse erlauben eine Bewertung der jeweiligen Architekturalternative sowohl von Software als auch von Hardware Komponenten.

### **Beitrag 2.1.2: Performanzanalyse für verteilte mikroelektronische Systeme unter Berücksichtigung von unterschiedlichen Kommunikationsprotokollen und Ereignismodellen**

Im Rahmen von Beitrag 2.1.2 wurden durch das FZI neue Methoden erforscht, welche eine Echtzeit- und Leistungsfähigkeitsanalyse verteilter eingebetteter Systeme unter Berücksichtigung der Eigenschaften der Verbindungsarchitektur, der eingesetzten Kommunikationsprotokolle, sowie des eingesetzten Softwareschedulings auf Verarbeitungseinheiten ermöglicht. Basierend auf der Vorgehensweise ist es möglich, ein ganzheitliches Modell eines eingebetteten mikroelektronischen Systems auf Performanzgrößen wie Auslastung oder aber Ende-zu-Ende-Latenzen zu prüfen und harte Echtzeitanforderungen zu verifizieren.

Der grundlegende Ansatz basiert dabei im Gegensatz zu anderen etablierten Ansätzen nicht auf einer explizit in der Analysedomäne modellierten Systemrepräsentation sondern auf der Extraktion des Analysemodells aus funktionalen Implementierungen von Systemteilen („White-Box“-Ansatz) und deren Komposition. Dieser Extraktionsansatz wurde in VISION um die modellbasierte Spezifikation, Komposition und Analyse der Verbindungsarchitektur, der Kommunikationsprotokolle, sowie des Softwareschedulings erweitert.

Der erweiterte Analyseablauf ist in Abbildung 26 dargestellt. Dabei repräsentieren gelbe Kästen neue Elemente des Spezifikations-/Analyseablaufs und gelb schattierte Kästen erweiterte Elemente.

Durch die erforschte Methodik wird ein Brückenschlag zwischen Echtzeitbewertung und modellbasiertem Systementwurfsprozess realisiert, welcher eine Anforderungsverifikation von verteilten eingebetteten Systems zu jedem Entwurfszeitpunkt unterstützt, bereits zu frühen Entwurfszeitpunkten die Verifikation von Echtzeiteigenschaften ermöglicht und somit frühzeitig Anforderungsverletzungen aufdecken hilft. Durch die Anwendung der erforschten Methoden kann potentiell die Anzahl von Entwurfsiterationen reduziert, die Iterationsdauer verkürzt und die dadurch verursachten Kosten verringert werden.

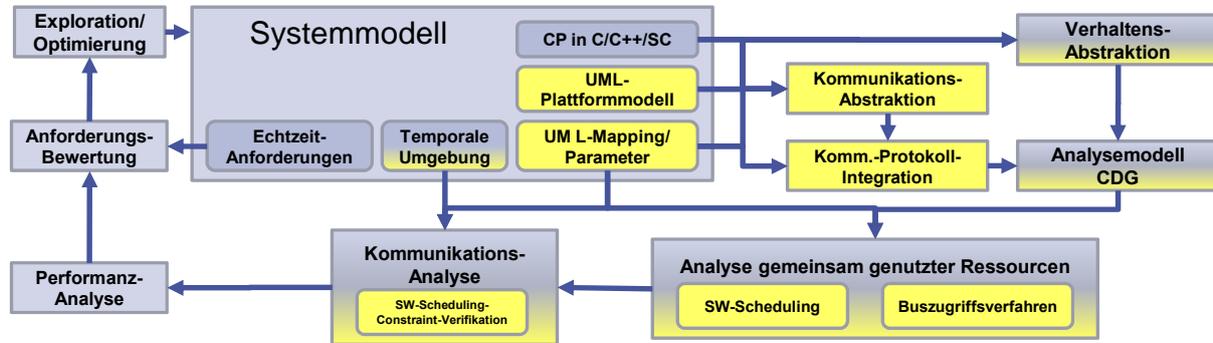


Abbildung 26: Erweiterter Analyseablauf

Die ersten Arbeiten umfassten die methodische Aufbereitung verschiedener Ereignis- und Umgebungsmodelle und deren Integration in das bestehende Verfahren zur kontrollflussorientierten Kommunikations- und Performanzanalyse. Hierfür wurden etablierte parametrisierbare Ereignismodelle untersucht und eine äquivalente Transformation in das Analysemodell durchgeführt. Dabei konnte gezeigt werden, dass ursprünglich für „Black-Box“-Analyseansätze entwickelte Ereignismodelle auch im „White-Box“-Analyseansatz anwendbar sind und damit die Einbeziehung eines Umgebungsmodells ermöglichen. Weiterhin wurde ein Ansatz zur Komposition verschiedener Ereignismodelle zur Berücksichtigung komplexer und reaktiver temporaler Umgebungen verfolgt. Diese Vorgehensweise ermöglicht die Verifikation der Eigenschaften von Systemmodellen unter Berücksichtigung der temporalen Systemumgebung.

Das Analysemodell wurde um Parameter zur Charakterisierung des Kommunikationsmediums wie Bandbreite und Latenz erweitert. Dadurch können statische Aussagen zum Systemverhalten in Abhängigkeit von der Kommunikationsarchitektur und der Applikation getroffen werden. Die Einbindung der definierten Modelle als parametrisierbare Schablonen von kommunizierenden Komponenten in das Modell wurde durchgeführt. Hierzu wurde ein Ansatz zur XML-basierten Repräsentation und Parametrisierung von Kommunikationsschablonen verfolgt. Dies ermöglicht die Anwendung des Analyseansatzes sowohl für eine gegebene modellbasierte Spezifikation als auch für einen in SystemC beschriebenen virtuellen Prototypen eines verteilten mikroelektronischen Systems, so dass eine abstraktionsebenenübergreifende Anwendbarkeit bereitgestellt werden kann. Die Ergebnisse dieses Beitrags bilden damit die Grundlage für die Arbeiten in Beitrag 1.2.3 zur Topologiefindung von verteilten Netzwerkarchitekturen.

Zur Repräsentation von Kommunikationskomponenten wird die in Aufgabe A1.1 entwickelte Methodik zur Systembeschreibung mit UML und IP-XACT verwendet. Dies ermöglicht eine standardisierte grafische Modellierung des Systems. Basierend auf einem UML-Systemmodell ermöglichen die eine sehr frühe Bewertung verschiedener Systementwurfalternativen, noch bevor ein realer oder virtueller Prototyp vorhanden sind. Durch eine enge Kooperation mit den Arbeiten zur Entwicklung einer Modellierungsmethodik konnte eine analysegerechte Aufbereitung der verschiedenen Systemaspekte (Plattform, Applikation, Abbildung, Parametrisierung) gewährleistet werden. Identifizierte Anforderungen, die notwendig für eine frühe Systembewertung sind, flossen zudem in die globale Modellierungsmethodik für mikroelektronische Systemverbände ein.

Aufgrund der Mächtigkeit der grafischen Modellierungssprache wurden Erweiterungen des Analysemodells vorgenommen, die eine Erhaltung der Semantik von für die Analyse notwendigen Aspekten gewährleisten. Hierzu mussten beispielsweise Komponentenübergreifende Verzweigungen im funktionalen Systemmodell auch im Analysemodell erhalten werden. Weiterhin mussten die Erfassung von Verhaltenshierarchien und deren Repräsentation im Analysemodell gewährleistet werden.

Zur Darstellung von Kommunikationsprotokollen auf höheren Ebenen wurde eine Verfeinerungs- und Analysemethodik entwickelt, die die Berücksichtigung des Einflusses auf die globale Performanz von verteilten, mikroelektronischen Systemverbänden erlaubt. Hierzu wurde ausgehend von Nutzdatenvolumina auf Applikationsebene im Analysemodell eine schichtenweise Integration und Verfeinerung der Kommunikation von einer Abstraktionsebene auf ISO/OSI auf die darunter liegende entwickelt. Die entwickelte Methodik zur automatisierten Überführung der Protokolle in das Analysemodell wurde ohne tiefgreifende Modellerweiterungen oder Abänderungen des grundlegenden Analyseablaufs entwickelt.

Weiterhin wurde die analyserechte Modellierung von Buszugriffsverfahren wie die der zentralen prioritätenbasierten Arbitrierung und der zeitscheibenbasierten Zugriffsverfahren untersucht. Zeitscheibenbasierte Verfahren sind in der adressierten Domäne der Automobilelektronik von besonderer Wichtigkeit, da sie Echtzeitgarantien erlauben und weit verbreitet sind.

Hinsichtlich der Berücksichtigung von Medienzugriffsverfahren wurde ein Analyseansatz entwickelt und umgesetzt, der die globale Leistungsfähigkeitsbewertung unter Berücksichtigung von statischen TDMA-basierten Zugriffsverfahren realisiert. Der Ansatz basiert auf einer Charakterisierung des Zugriffsverfahrens im Spezifikationsmodell und setzt die Parameter (z.B. Zeitscheibenlänge, Zuweisung von Prozessen zu Zeitscheiben) im formalen Analyseframework um. Es wurden Methoden zur pessimistischen best-case/worst-case Ermittlung des resultierenden Zeitverhaltens entwickelt und in das Framework integriert.

Die Weiterentwicklung des Analyseverfahrens umfasst zudem die Inklusion von dynamischen Segmenten mit Round-Robin-Strategien. Die Kombination von TDMA und Round-Robin erlauben die Bewertung applikationsrelevanter Bussystemmodelle, wie beispielsweise das Zeitverhalten des FlexRay-Busses.

Weiterhin wurde ein Ansatz zur Modellierung von kooperativem und nicht-präemptiven, sowie zur Modellierung von zeitscheibenbasiertem präemptiven Software-Ablaufplänen erforscht. Grundlegende Schritte hierzu waren die Integration der Repräsentation verschiedener Software-Ablaufpläne im Spezifikationsmodell und die Erweiterung des darunter liegenden formalen Analysemodell.

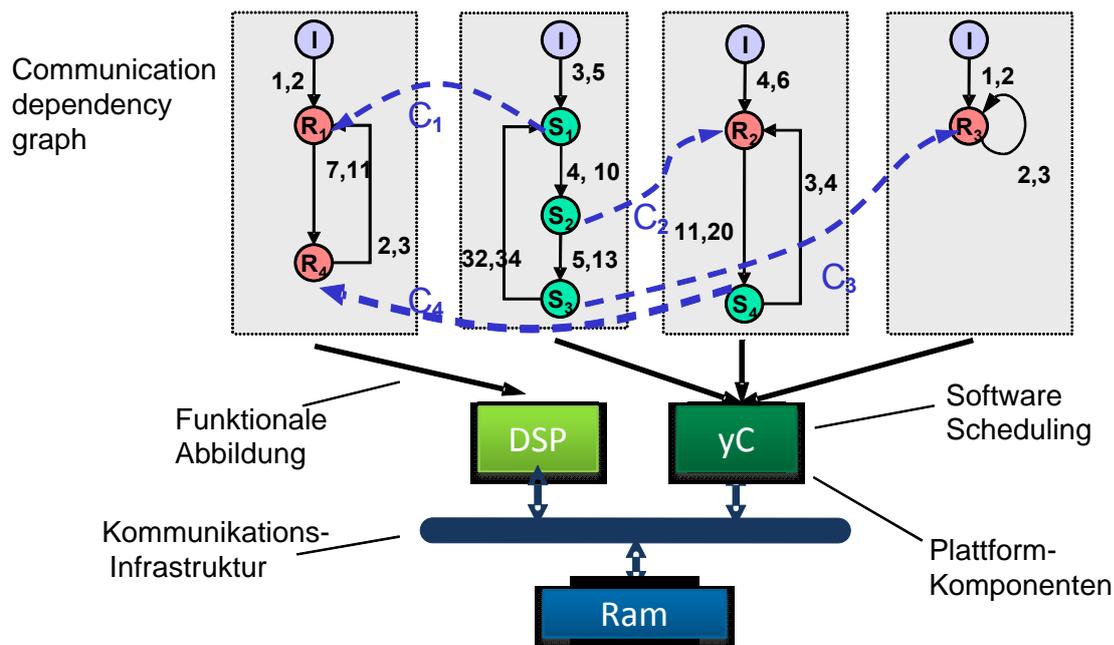


Abbildung 27: Parametrisierung der Abbildung kommunizierender Prozesse

Darüber hinaus wurden im Analysemodell Erweiterungen mit dem Ziel der Repräsentation hierarchischen Verhaltens spezifiziert. Die Erweiterungen charakterisieren Unterbrechungspunkte im Kontrollfluss der kommunizierenden Prozesse, sowie die Aktivierung-/Deaktivierungs-Sensitivität von Prozessen auf Unterbrechungen. Es wurden Einschränkungen auf dem Modell spezifiziert, welche eine Klassifizierung der verschiedenen Ablaufpläne erlauben und eine Analysierbarkeit der Modellerweiterungen sicher stellen.

Zeitliche Eigenschaften wie Interruptlatenzen, und Kontextwechselzeiten wurden definiert und ein Analysealgorithmus entwickelt, welcher unter Berücksichtigung der Modellerweiterungen eine Bewertung von Systemmodellen mit Software-Scheduling erlaubt. Dabei können verschiedene Strategien wie kooperatives Scheduling, statisches Reihenfolgescheduling, nicht-präemptives Zeitscheibenscheduling, aber auch die explizite Aktivierung von Prozessen auf verschiedenen Verarbeitungseinheiten modelliert und parametrisiert werden. Darüber hinaus wurde die für die Berücksichtigung von TDMA-Zugriffsverfahren auf Kommunikationsressourcen entwickelte Analysemethodik auf spezifische Parameter von Softwarescheduling wie Kontextwechselzeiten und Verdrängungskosten adaptiert und eine Untersuchung dynamischer Aspekte integriert. Damit lassen sich kombinierte TDMA-/Round-Robin-Analysen für Softwarescheduling analysieren. Darüber hinaus wurde gezeigt, dass verschiedene Schedulingverfahren wie statisch-kooperatives Scheduling und TDMA-Scheduling durch eine hierarchische Vorgehensweise analysiert werden können. Zudem erlaubt die kompositionelle Vorgehensweise die gleichzeitige Betrachtung von Busprotokollen und Task-Verdrängung.

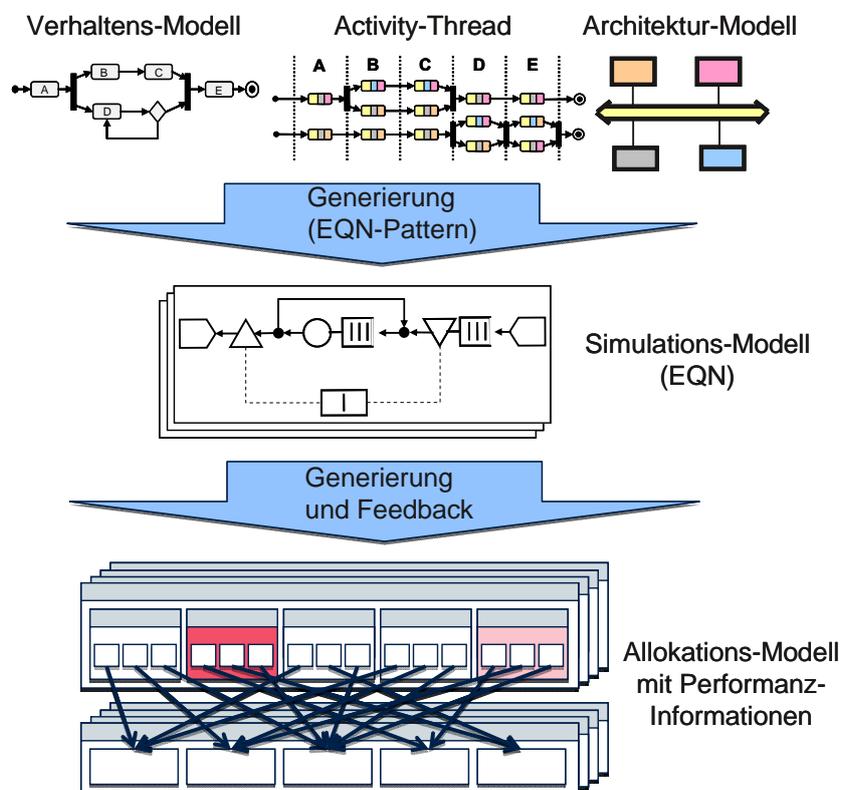
In Abbildung 27 ist die aus den Analyseerweiterungen resultierende Abbildung von kommunizierenden Prozessen auf Plattformkomponenten dargestellt. Die der im Gesamtprojekt entwickelten modellbasierten Vorgehensweise zum modellbasierten Entwurf verteilter mikroelektronische Systeme angelehnte Analysemethodik unterstützt durch die in VISION erforschten Erweiterungen die Beschreibung der Kommunikationsinfrastruktur (Kommunikationsprotokolle und Zugriffsmethoden), die funktionale Abbildung, die Charakterisierung der Plattformkomponenten, die Verwendung von Software-Scheduling, sowie die Einbeziehung formal spezifizierter Umgebungsmodelle.

Die entwickelten Verfahren wurden prototypisch implementiert und in das SysXplorer-Werkzeug integriert.

### Beitrag 2.1.3: Methodik zur Performanzanalyse von verteilten Systemen basierend auf einer abstrakten Systembeschreibung.

In diesem Arbeitspaket wurde von Infineon eine Methodik zur Performanzanalyse basierend auf abstrakten Systembeschreibungen entwickelt. Für die Systembeschreibungen wurde MARTE-UML verwendet, ein UML-Profil zur Beschreibung von Echtzeit- und Eingebetteten Systemen. Aus diesen Beschreibungen wurde ein durch Erweiterte Warteschlangennetze (EQN) beschriebenes System generiert, simuliert und ausgewertet.

Der entwickelte Ansatz eignet sich besonders zur Evaluation von Designalternativen hinsichtlich der zur erwartenden Performanz des Gesamtsystems. Zur Beschreibung dieser Designalternativen wurden sogenannte Activity-Threads eingeführt. Diese werden, wie in Abbildung 28 zu sehen, zusammen mit Beschreibungen der Architektur und dem Verhalten des Systems definiert.



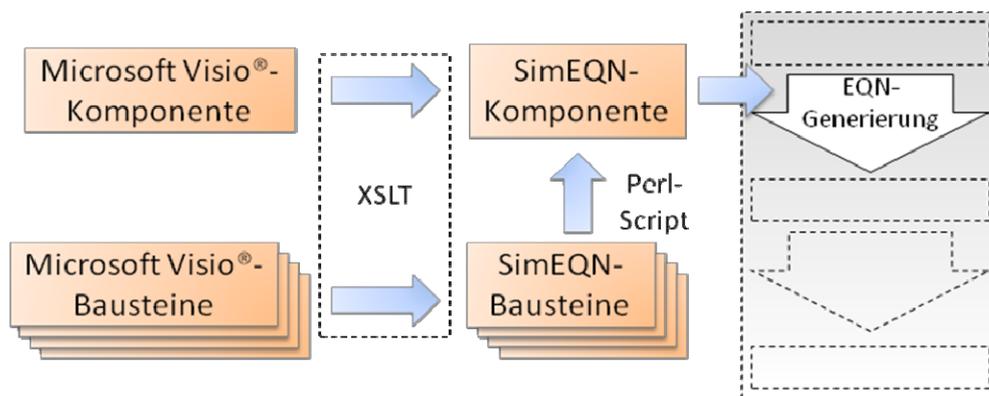
**Abbildung 28: Framework zur Generierung und Auswertung von Systemmodellen zur Performanzanalyse**

Das Verhaltens-Modell, welches die Funktionalität des zu entwickelnden Systems repräsentiert, wird unter Verwendung von Aktivitäts-Diagrammen spezifiziert. Um performanzrelevante Angaben an die Aktivitäten des Diagramms annotieren zu können, kommt das UML Real Time Execution Model of Computation and Communication (RTEMoCC), ein Bestandteil des MARTE Profils, zum Einsatz. Dieses erweitert das ursprüngliche UML Meta-Modell für Aktivitäts-Diagramme um Annotationsmöglichkeiten zur Spezifikation von Echtzeit-Anforderungen. Die Granularität des verwendeten Funktionalitäts-Modells wird dabei so gewählt, dass jeder Aktivität eindeutig eine Anzahl an Ressourcen zugeordnet werden kann, die während der Ausführung der spezifizierten Funktion teilweise oder komplett allokiert werden.

Das Architektur-Modell wird durch ein UML Kompositions-Struktur-Diagramm spezifiziert und definiert die zur Verfügung stehenden System-Ressourcen. Dieses Modell wird mit Stereotypen aus dem MARTE Hardware Ressource Model (HRM) zur hardware-seitigen Beschreibung der Ausführungs-Plattform informationstechnischer Systeme erweitert. Hierbei wird der Bezug zum zu generierenden Simulationsmodell mittels Verknüpfungen zu einer Komponenten-Datenbank hergestellt, in der komponenten-spezifische Pattern hinterlegt sind, aus denen das EQN-Modell komponiert werden kann. Weiterhin enthält das Architektur-Modell Informationen über die Kommunikations-Strukturen zwischen den zum Einsatz kommenden Architektur-Komponenten.

Ein Activity-Thread spezifiziert, welche Funktionalität auf welchen Hardware-Komponenten ausgeführt wird. Dabei wird es durch die Nutzung eines Activity-Threads ermöglicht, mittels eines Mapping-Modells alternative System-Implementierungen zu spezifizieren, welche anschließend automatisiert hinsichtlich ihrer Performanz bewertet werden können. Zur Modellierung des Activity-Threads kommt dabei ein UML Aktivitäts-Diagramm zum Einsatz, welches jedoch mit einer alternativen Semantik belegt wird. Jeder Pfad durch das Diagramm repräsentiert eine zu untersuchende Systemvariante. Dabei entspricht jeweils eine Aktivitäts-Partition eines Activity-Threads einer Aktivität des Verhaltens-Modells. Die Aktivität im Activity-Thread spezifiziert wiederum die zur Ausführung dieses Bestandteils der Funktionalität zum Einsatz kommenden Hardware-Komponenten. Da der Activity-Thread Verzweigungen zulässt, lässt sich so der zu Untersuchende Design-Space manuell auf die jeweils sinnvollen Systemalternativen reduzieren.

Im Anschluss wird für jede modellierte Systemalternative ein Simulationsmodell erzeugt und ausgeführt. Zur Erzeugung eines Systemmodells werden für die verschiedenen in der Architekturbeschreibung verwendeten Hardware-Typen Modell-Bausteine benötigt. Diese lassen sich direkt in XML oder durch die Verwendung von Microsoft Visio® erzeugen.



**Abbildung 29: Erzeugung von Komponenten-Modellen**

Für die Generierung dieser Templates gibt es zwei Verfahren, abhängig davon ob ein statisches oder dynamisches Modell verwendet werden muss. Statische Modelle können durch in sich geschlossene EQNs mit einem Eingang und einem Ausgang repräsentiert werden. Diese EQNs lassen sich anschließend mittels Template-Variablen aus den UML-Beschreibungen konfigurieren. Dieses Prinzip lässt sich z.B. auf Speicher, Prozessoren und andere Datenverarbeitende Komponenten anwenden. Dynamische Modelle hingegen müssen aus verschiedenen Bausteinen mittels eines Hilfsprogramms, z.B. eines Perl-Scripts, zusammengesetzt werden. Die Struktur der neuen Komponente hängt dabei von durch den Anwender zu spezifizierenden Template-Variablen ab. Dieses Prinzip ist beispielsweise bei Bus-Komponenten anzuwenden die eine beliebige Anzahl von Verbindungen beinhalten können.

Wie in Abbildung 29 dargestellt, können sowohl einzelne Komponenten-Modelle, als auch die für die dynamischen Modelle notwendigen Bausteine mittels Microsoft-Visio® generiert

werden. Um eine von der grafischen Darstellung unabhängige Beschreibung der EQNs zu erhalten werden diese mittels einer XSL-Transformation in eine simulierbare XML-Beschreibung transformiert.

Die nach der Simulation gewonnenen Erkenntnisse über die zu erwartende System-Performanz werden in ein MARTE Allokationsmodell eingefügt und dem Anwender in seiner UML Modellierungsumgebung verfügbar gemacht. Dazu wurde das Meta-Modell für MARTE Allokationsmodelle um einen Stereotypen zur Visualisierung von Performanz erweitert.

## Aufgabe 2.2: Überprüfung applikationsspezifischer Anforderungen

### Beitrag 2.2.1: Methodik zur Erfassung komplexer Umgebungsbedingungen des Systems und Anforderungen an das System sowie deren Integration in den Systementwurfsablauf

Im Rahmen des Beitrags 2.2.1 wurden von Bosch Methoden erforscht, die eine Erfassung von vielschichtigen Randbedingungen von verteilten Systemen ermöglichen und es so erlauben, komplexe Systemeigenschaften wie etwa Zuverlässigkeit, Sicherheit, Echtzeit- und Fehlertoleranzverhalten zu berücksichtigen. Basierend auf diesen Methoden wird eine Vorgehensweise aufgebaut, die es ermöglicht, komplizierte Umgebungs- und Randbedingungen des Systems in das abstrakte Systemmodell zu integrieren und auf diese Weise dem nachfolgenden Systementwurfsprozess zuzuführen.

In den Arbeiten von Bosch in Beitrag 2.2.1 erfolgte als Beispiel für umfassende Systemanforderungen eine Fokussierung auf die Systemeigenschaften Safety und Security sowie auf die für komplexe Netzwerke relevante Fragestellung einer Kopplung von Subnetzdomänen mit unterschiedlichen Charakteristika zu einem heterogenen Gesamtnetzwerk. Hierzu wurde als Lösung eine Gateway-Struktur mit „Firewall-Funktionalität“ konzipiert und als Realisierungsansatz die in Abbildung 30 dargestellte Hardware-Architektur bestehend aus einem Multiprozessorsystem mit spezialisiertem Coprozessor („Gateway Control Unit“) vorgeschlagen.

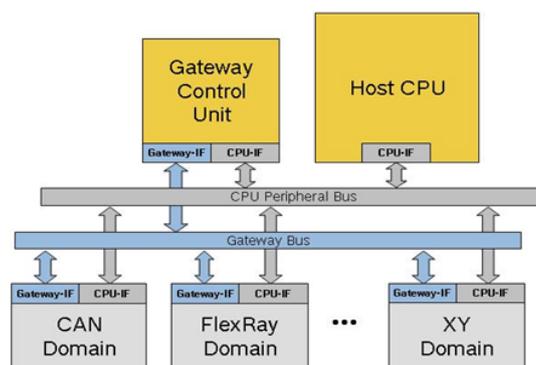
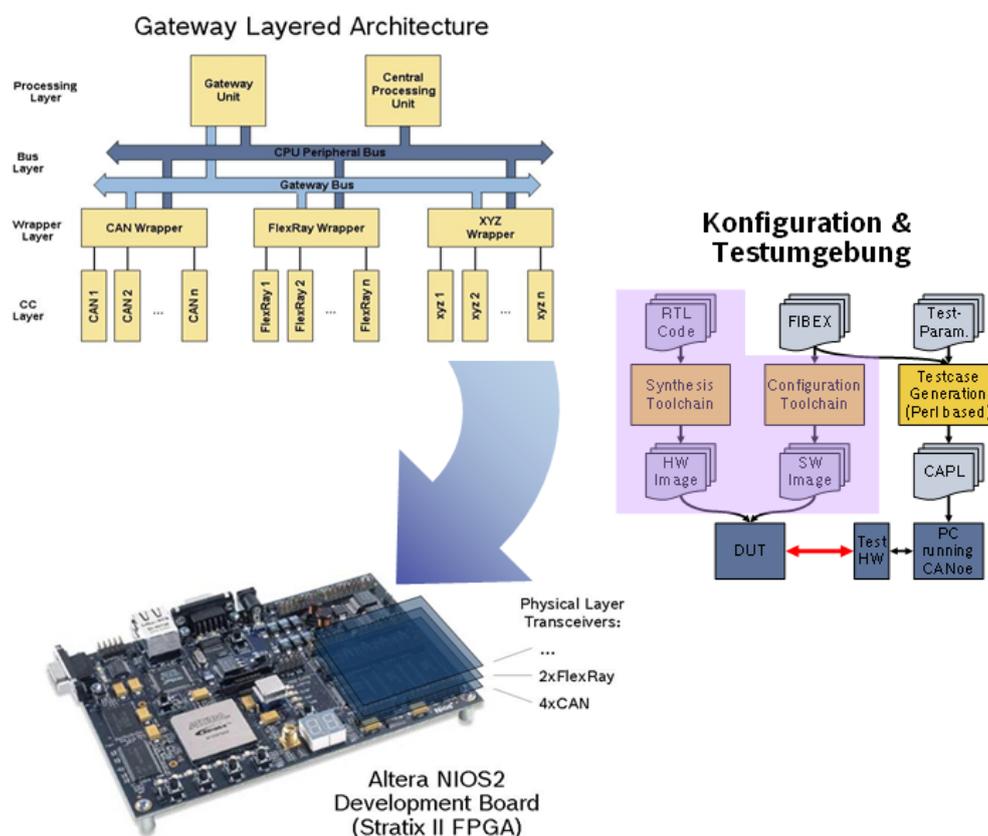


Abbildung 30: Gateway-Architektur zur Kopplung von Subnetzdomänen

Für den vorgeschlagenen Lösungsansatz erfolgte eine Umsetzung in einen Hardware-Prototyp auf FPGA-Basis. Dieser ermöglichte eine detaillierte Validierung und Analyse des Vorgehens. Hierzu wurden prototypische Werkzeuglösungen realisiert, die eine automatisierte Konfiguration des Hardware-Prototyps ermöglichen sowie eine automatisierte Testumgebung für Gateways bereitstellen, die einen Black-Box-Test beliebiger Gatewaystrukturen sowie den automatischen Vergleich ihrer Soll-/Ist-Funktionalität unterstützen. Auf diese Weise wurde eine flexible Umgebung für eine Hardware-in-the-Loop Simulation von konfigurierbaren Architekturen auf FPGA-Basis geschaffen, die für eine detaillierte Untersuchung und Explo-

ration des Konzepts hinsichtlich Safety- und Security-Eigenschaften eingesetzt werden konnte. Abbildung 31 zeigt einen Überblick über die Konfigurations- und Testumgebung und das Vorgehen.

Aufbauend auf der entwickelten Multiprozessor-Architektur zur Realisierung von Sicherheitsmechanismen sowie der entstandenen flexiblen Konfigurations- und Testumgebung für Gateway-Varianten auf FPGA-Basis erfolgte in einem weiteren Schritt die Anpassung des Konzepts an aktuelle und zukünftige Anforderungen und Randbedingungen des Automotive-Systementwurfsprozesses. Dabei wurden insbesondere die zunehmende Verschiebung der Hardware/Software-Partitionierung hin zu Software-lastigeren Systemen sowie die Unterstützung aktueller Standards wie Autosar in die Betrachtungen einbezogen. Auf Basis dieser erweiterten Anforderungen erfolgte eine Neukonzeption eines Kommunikationscontrollers, im betrachteten Fall auf Basis eines CAN Controllers. Im Rahmen der Arbeiten wurde die Struktur eines neuartigen CAN-Controllers (M\_CAN) konzipiert, das Konzept sowie applikationsnahe Treiber wurden implementiert und deren Integration in eine System wurde anhand einer typischen Steuergeräteumgebung auf FPGA-Basis untersucht.



**Abbildung 31: Simulations- und Analyseumgebung**

Abbildung 32 zeigt in (a) die Architektur des M\_CAN Controllers. Diese beinhaltet als im Kontext der genannten Anforderungen wichtige Komponenten erweiterte Tx- und Rx-Einheiten, welche die Nachrichtenpriorisierung beim Nachrichtentransfer vom externen Nachrichtenspeicher zum CAN-Core (Tx-Handler) bzw. die Akzeptanzfilterung beim Nachrichtentransfer vom CAN-Core zum externen Nachrichtenspeicher (Rx-Handler) kontrollieren. Abbildung 32 (b) zeigt ein Beispielsystem mit drei M\_CAN Instanzen und einem gemeinsamen Nachrichtenspeicherbereich im Systemspeicher.

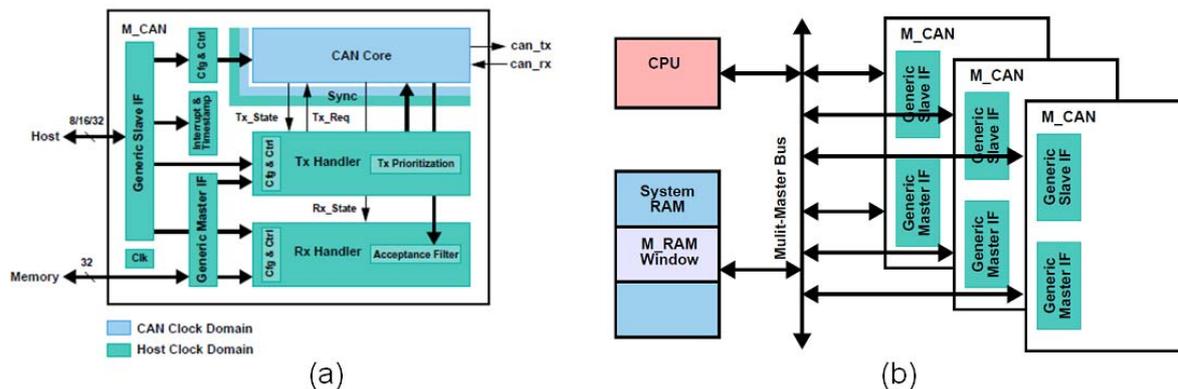


Abbildung 32: M\_CAN Controller Architektur (a) und Systemeinbindung (b)

Die Arbeiten von Bosch in Beitrag 2.2.1 wurden durch den Unterauftragnehmer OFFIS unterstützt, dessen Schwerpunkt die Anbindung von Prototypen an die Hardware-Verifikation und den Hardware-Test auf Basis des in Beitrag 3.1.1 entwickelten SystemC-Verifikationsansatzes darstellte.

### 4.3 AP3: Verifikationsgestützte Systemintegration und –implementierung

Während Arbeitspaket 1 und 2 eine Modellierungs- und Analyseumgebung für verteilte Systeme adressieren, schloss Arbeitspaket 3 die Lücke zum nachfolgenden Implementierungsprozess. Hierzu wurden Verfahren erforscht, die im weiteren Entwurfsablauf eine Verifikation der Systemintegration und einen Entwurf des verteilten Systems über Abstraktionsebenen und Domänengrenzen hinweg unterstützen und so den Prozess der Systemimplementierung ganzheitlich begleiten. Dabei nahm der Einsatz von standardisierten Subsystemen und Schnittstellen einen zentralen Stellenwert ein. An dieser Stelle definiert das verteilte Szenario Randbedingungen, etwa im Hinblick auf Flexibilität bezüglich neuer Kommunikationsstandards und -normen oder im Hinblick auf Sicherheits- und Zuverlässigkeitseigenschaften, die in bisherigen Entwurfs- und Verifikationsmethoden nicht hinreichend berücksichtigt wurden. Die in Arbeitspaket 3 entwickelten Verfahren adressieren erstmals eine Umsetzung dieser spezifischen Randbedingungen, wobei sowohl seitens des Entwurfs als auch seitens der Verifikation Wiederverwendungsparadigmen eine wichtige Rolle einnehmen.

Durch methodische Ansätze zur automatisierten Umsetzung abstrakter Systemmodelle in implementierbare Entwurfsbeschreibungen und deren Verifikation gegen das Systemmodell konnte der Zeitpunkt der Systemintegration auf Modellebene durchgeführt und so innerhalb des Produktentstehungsprozesses nach vorne verschoben werden. Auf diese Weise wurde die Systemsicherheit erhöht und die Entwurfszeit reduziert. Dies ermöglicht es, die hohen Anforderungen an Flexibilität und Qualität zukünftiger vernetzter mikroelektronischer Systeme mit wirtschaftlich vertretbarem Aufwand zu realisieren. Dabei kann die ganzheitliche Sicht auf das verteilte System sowie dessen Umgebungseigenschaften nahtlos in den Verifikations- und Implementierungsprozess integriert werden. Die methodische Beherrschung der genannten Anforderungen liefert ein Alleinstellungsmerkmal, das den Aufbau neuer innovativer verteilter Systemlösungen ermöglicht.

### Aufgabe 3.1: Verifikation der Systemintegration vernetzter Architekturen

#### Beitrag 3.1.1: Entwicklung von Methoden zur modularen Verifikation von Systemkomponenten und deren Integration in ein vernetztes mikroelektronisches System

Im Rahmen des Beitrags 3.1.1 wurden von Bosch Methoden untersucht, die in modularer Weise eine Verifikation komplexer Systemkomponenten sowie deren Integration in ein vernetztes Gesamtsystem ermöglichen. Die adäquate Berücksichtigung des Aspekts der Wiederverwendung von verteilten Systemkomponenten (Schnittstelle zu Beitrag 3.2.1) und Verifikationsmodulen war ein wichtiger Schwerpunkt. Ein weiterer Schwerpunkt war die Bereitstellung von Mechanismen zur zentralen Steuerung und Synchronisation dezentraler Verifikationseinheiten. Letzteres bildet die Grundlage für die Abbildung komplexer Anforderungen und Randbedingungen eines verteilten Systems in den Verifikationsprozess (Schnittstelle zu Beitrag 2.2.1).

Hierzu wurde von Bosch eine modulare und wiederverwendbare SystemC-basierte Methodik zur Verifikation von Systemmodulen und deren Integration in ein vernetztes Gesamtsystem entwickelt und implementiert. Der Ansatz basiert auf der Systembeschreibungssprache SystemC und deren Erweiterung (Zusatzbibliothek) SCV (SystemC Verification Library). Abbildung 33 liefert einen Überblick über das Vorgehen: Jeder Schnittstelle des zu verifizierenden Designs (Design under Test, DUT) wird *ein* Verifikationsmodul (Testbench-Modul, TM) zugeordnet. Eine strukturelle Verdrahtung erfolgt *ausschließlich* zwischen Verifikationsmodulen und DUT. Eine Steuerung des Verifikationsablaufs erfolgt durch *eine* Steuereinheit (Testbench-Controller), welche durch eine zentrale Kommando-Datei mit Schnittstellen- und Synchronisationskommandos für die beteiligten Verifikationsmodule gespeist wird. Eine Anbindung der Verifikationsmodule an die Steuereinheit erfolgt über eine virtuelle Verdrahtungsebene.

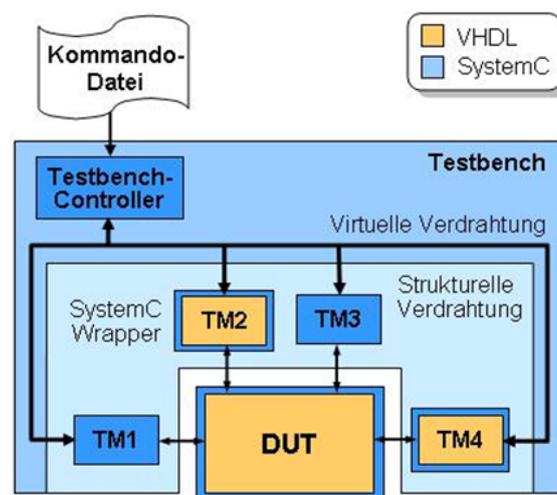


Abbildung 33: SystemC-basierte Verifikationsmethodik

Als wichtige Anforderung für eine Einführung der entwickelten Methodik in den industriellen Entwurfsablauf bei Bosch wurde dabei auf eine nahtlose Integration des neuen Vorgehens in eine bestehende (und im industriellen Entwurfsablauf von Bosch etablierte) VHDL-basierte Verifikationsstrategie geachtet. Hierzu wurde ein Co-Simulationsansatz realisiert, der eine kombinierte Anwendung von bestehenden (VHDL-basierten) und neuen (SystemC-basierten) Verifikationsmodulen unterstützt. Auf diese Weise können über die neuen Verifikationsmodule erweiterter Verifikationsmechanismen zur Verfügung gestellt werden, ohne dabei die Anwendbarkeit existierender Verifikationskomponenten zu gefährden.

Das entwickelte Verifikationskonzept wurde im weiteren Verlauf der Arbeiten in Beitrag 3.1.1 hinsichtlich der Bereitstellung fortschrittlicher Verifikationsmechanismen und der Verbesse-

Die Erweiterung der Anwendbarkeit im Kontext vernetzter (Automotive-)System szenarien erweitert. Diese Erweiterungen beinhalteten den Aufbau und die Integration von Mechanismen zur Randomisierung von Testdaten und Verifikationsablauf sowie Ansätze für ein interaktives „Debugging“. Einen weiteren Arbeitsschwerpunkt stellte die Erweiterung des Vorgehens um einen Ansatz zur flexiblen Integration von Applikationssoftware in den Verifikationsablauf dar, Abbildung 34 zeigt einen Überblick. Auf diese Weise konnte das bisherige Vorgehen einer Simulation der Software auf Maschinencodeebene auf einem zyklengenauen Modell des Zielprozessors durch eine *direkte* Integration des Applikationscodes auf Sourcecode-Ebene abgelöst und auf diese Weise die Performanz und Flexibilität der Software-Anbindung deutlich erhöht werden. Diese Option besitzt insbesondere vor dem Hintergrund eines steigenden Anteils von Software in automobilelektronischen Netzwerken eine hohe Bedeutung.

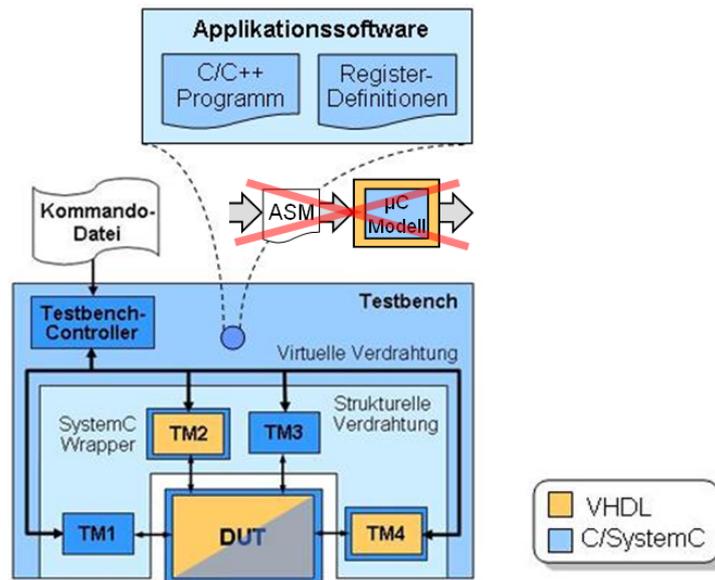


Abbildung 34: Anbindung von Applikationssoftware

Weitere Schwerpunkte bildeten die Konzeption und der Ausbau der Verifikationsumgebung um Mechanismen zur flexiblen (zentralisierten) Steuerung und Synchronisation des (dezentralen) Verifikationsgeschehens. Hierzu wurde eine flexible und auf die konkrete Anwendung zugeschnittene Skriptsprache definiert und diese in Kombination mit leistungsfähigen Scripting-Mechanismen innerhalb der Verifikationsumgebung zur Verfügung gestellt. Last but not least wurde in enger Kooperation mit dem Projekt VeronA eine Erweiterung des Konzepts für eine Anwendung mit VHDL-AMS und damit für einen Einsatz in Analog-/Mixed-Signal-Szenarien vorgenommen und prototypisch implementiert.

Die entwickelte SystemC-basierte Verifikationsmethodik unterstützt zum einen eine Anwendung auf die innerhalb des VISION-Designflows zentrale Datenstruktur des virtuellen Prototypen und ermöglicht zum anderen die Anwendung *derselben* Verifikationsumgebung innerhalb des nachfolgenden VHDL-basierten Verifikationsablaufs – und liefert somit ein Bindeglied zwischen virtuellem Prototyp und Implementierungsprozess.

Die Arbeiten von Bosch in Beitrag 3.1.1 wurden durch den Unterauftragnehmer OFFIS durch enge Zusammenarbeit bei der Implementierung der Umgebung, beim Aufbau von Verifikationsmodulen und bei der prototypischen Anwendung der entwickelten Mechanismen unterstützt.



---

**Beitrag 3.1.3: Erstellung eines „Layered Modelling Approach“ für verifizierbare virtuelle Prototypen für verschiedene Anwendungen.**

Im Beitrag 3.1.3 adressierte Infineon die Entwicklung von Komponenten für den sogenannten virtuellen Prototypen. Ein virtueller Prototyp ist ein ausführbares Architekturmodell, das als Referenz für weitere Entwicklungsschritte verwendet werden kann. Dies bietet insbesondere für den Entwurf von Hardware/Software Systemen eine erhebliche Verkürzung der Time-to-Market, da mit einigen Entwurfsschritten – z.B. dem Software-Entwurf – bereits begonnen werden kann, bevor die Hardware vorliegt.

Die Entwicklung des virtuellen Prototypen geschieht auf mehreren Abstraktionsebenen, wobei jede Entwicklungsstufe eine Verfeinerung der vorhergehenden darstellt. Ebenso hat jede Stufe eine Anwendung im Entwurfsprozess.

In dem Projektbeitrag 3.1.3 wurde nun eine Methodik ("Layered Modelling Approach") entwickelt, mit der die Modelle der entsprechenden Ebenen in der nächsten Ebene ohne Änderung wiederverwendet werden können. Dies bietet die folgenden Vorteile für den Entwurf des virtuellen Prototypen:

- Zeitersparnis bei dem Entwurf, da die Modelle der vorherigen Ebenen nicht nochmal neu implementiert werden müssen.
- Erweiterte Verifikationsmöglichkeiten bei der Verfeinerung der Modelle. Ist das gesamte System (der virtuelle Prototyp) in einer Ebene aufgebaut, so kann das verfeinerte System der nächsten Ebene komponentenweise entwickelt werden. Dies ist möglich, da mittels der "Layered Modelling" Methodik das bestehende System als Testumgebung für die neu zu erstellende Komponente benutzt werden kann.

Im ersten Schritt wurden zunächst die sinnvollen Abstraktionsebenen für die Modelle identifiziert. Die nachfolgenden Arbeiten konzentrierten sich demnach auf die folgenden drei Modellierungsebenen:

- Funktionale Ebene. Die Modelle auf dieser Ebene weisen nur die funktionalen Eigenschaften der Komponente ohne zeitliches Verhalten auf. Mit dieser Stufe des virtuellen Prototypen wird die Korrektheit der entwickelten Algorithmen bzw. des Datenpfades überprüft.
- Transaktionsebene (auch "TLM-PVT" für "Transaction Level Model – Programmer's View with Timing"). Auf dieser Ebene wird zeitliches Verhalten bereits grob modelliert, allerdings wird von komplexen Kommunikationsprotokollen abstrahiert. Der gesamte Transfer eines Datenpakets zwischen Komponenten des Systems wird als eine abstrakte "Transaktion" modelliert. Diese Stufe des virtuellen Prototyps kann zur Softwareentwicklung herangezogen werden, bevor das System als physikalische Hardware existiert.
- Signalebene. Auf dieser Ebene werden auch die Kommunikationsprotokolle modelliert. Diese Modelle können als Referenz für die RTL-Entwicklung verwendet werden.

Für diese drei Ebenen wurde eine konkrete Methodik für die Implementierung dieser Modelltypen entwickelt, die die Wiederverwendung der Modelle der höheren Ebenen in den detaillierteren Ebenen erlaubt. Hierdurch ergibt sich ein Schichtenmodell ("Layered Model"), bei dem der Verfeinerungsschritt durch das Hinzufügen zusätzlichen Codes zu der bestehenden Schicht erreicht wird. Das Schichtenmodell ist in Abbildung 36 illustriert.

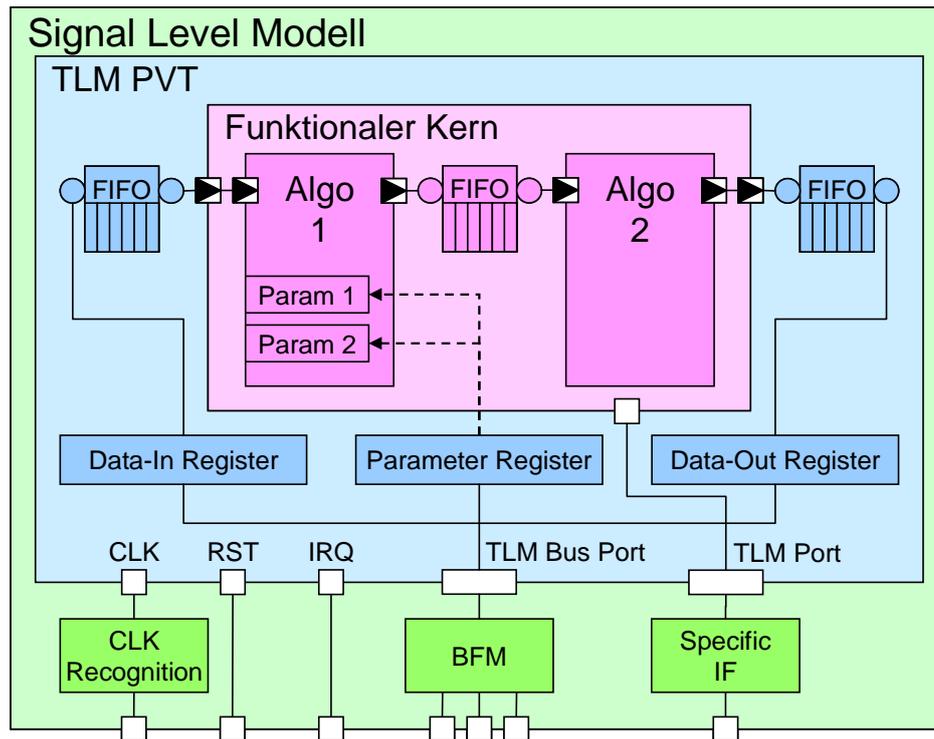


Abbildung 36: Schichten des "Layered Modelling Approach"

Im zweiten Schritt wurde die Methodik für konkrete Komponenten angewandt, wobei besonderer Fokus auf die Transaktionsebene und die Signalebene gelegt wurde. Der Übergang zwischen diesen beiden Ebenen ist besonders kritisch, da hierbei ein besonders detailliertes Zeitverhalten zu dem Modell hinzugefügt wird. Konkret müssen hierbei (vgl. Abbildung 36) die Behandlung des RTL-Taktes (Clock Recognition), die Umsetzung des Busprotokolls (BFM – "Bus Functional Model") sowie eine spezifische Anpassung für protokolle von modul-spezifischen Schnittstellen implementiert werden.

Im dritten Schritt wurde der entwickelte Ansatz schliesslich auf ein Beispiel angewandt, wobei der oben besprochene Anwendungsfall der Einbettung eines Moduls auf Signalebene in ein System auf Transaktionsebene modelliert wurde. Dies geschieht durch Einsetzen von entsprechenden Transaktoren, die die Umsetzung der Protokolle durchführen. Durch den "Layered Modelling Approach" konnte dabei die notwendige Verifikation auf den Übergang von TLM auf Signalebene beschränkt werden. Hierzu wurden zusätzliche Monitoring-Anschlüsse und Komparatoren implementiert. Der Gesamtaufbau ist in Abbildung 37 dargestellt.

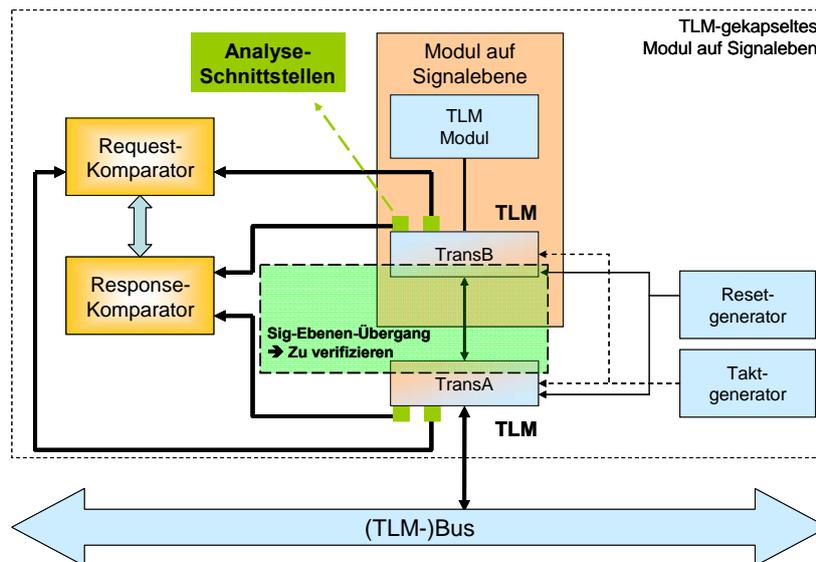


Abbildung 37: Anwendung des "Layered Modelling Approach"

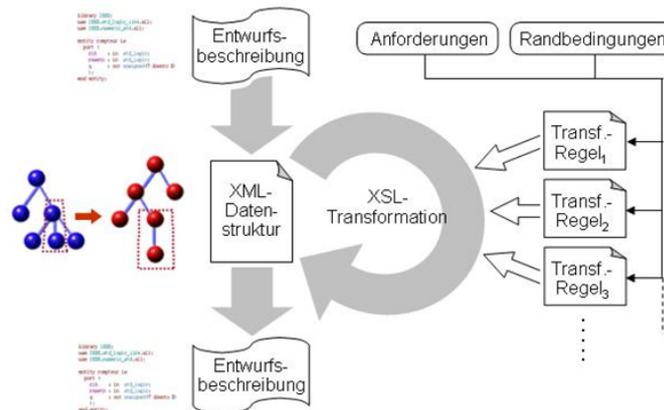
Das Ergebnis des Beitrags ist somit eine effiziente Methodik für die Entwicklung von virtuellen Prototypen auf verschiedenen Ebenen, bei der Aufwand durch Wiederverwendung bereits existierender Schichten gespart wird. Die Methodik erlaubt die Einbettung von neu zu entwickelnden Komponenten einer Ebene in das bestehende System einer anderen Ebene. Dies funktioniert sowohl bei einer detaillierteren Komponente in einem abstrakteren System (vgl. Abbildung 37) als auch umgekehrt.

### Aufgabe 3.2: Durchgängige Implementierungsprozesse für verteilte Systemlösungen

#### Beitrag 3.2.1: Entwicklung von Methoden zur zielgerichteten Verbesserung von kritischen Systemeigenschaften in einer vernetzten Systemumgebung

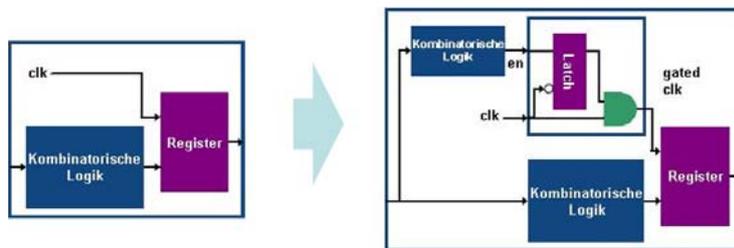
Im Rahmen des Beitrags 3.2.1 wurden von Bosch Methoden erforscht, die es ermöglichen, die charakteristischen Eigenschaften eines vernetzten Systems zielgerichtet und in automatisierter Weise zu beeinflussen. Die entwickelten Methoden adressieren zum einen einzelne Systemkomponenten der verteilten Systemumgebung, unterstützen zum anderen aber auch eine ganzheitliche Erfassung des Netzwerkverbands.

Hierzu wurde von Bosch ein Ansatz zur flexiblen zielgerichteten Manipulation von Systembeschreibungen auf Basis von XML konzipiert. Dabei erfolgt eine Umsetzung einer Systembeschreibung in eine XML-Darstellung. Diese liefert die Grundlage für die Anwendung von Manipulationsverfahren, die in Form von wiederverwendbaren XSLT-Transformationsregeln gegeben sind. Auf diese Weise können leistungsfähige XSLT-Prozessoren zur Durchführung der Transformation eingesetzt werden. Im Anschluss daran kann die transformierte XML-Darstellung zur weiteren Verarbeitung im Systementwurfsablauf zurück in eine Systembeschreibung transformiert werden. Abbildung 38 zeigt einen Überblick über das Vorgehen. Der entwickelte Ansatz zeichnet sich durch eine hohe Flexibilität hinsichtlich der Art und Gestaltung von Transformationsregeln aus und ist in der Lage, ein breites Spektrum von Transformationen zu unterstützen.



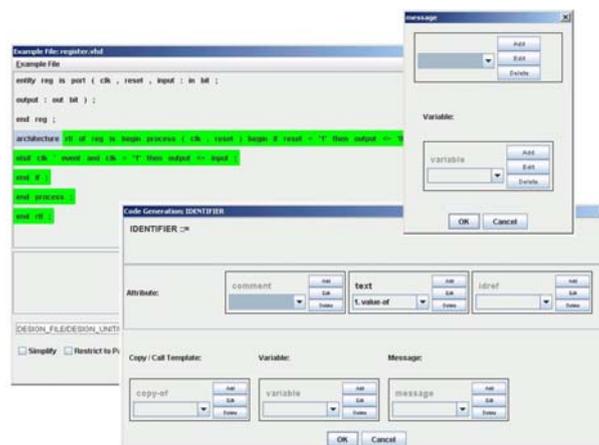
**Abbildung 38: XML/XSLT-basierter Transformationsansatz**

Die Leistungsfähigkeit des entwickelten Ansatzes wurde durch die Realisierung einer Transformation zur Reduktion des Leistungsverbrauchs einer Schaltung nachgewiesen. Hierzu erfolgte in Kooperation mit dem Projekt LEMOS der Aufbau einer Transformationsregel zum Einfügen von Clock Gates innerhalb einer RTL-Beschreibung gemäß Abbildung 39.



**Abbildung 39: Low-Power Transformation basierend auf Clock-Gating**

Zur Erhöhung der Anwendbarkeit des Ansatzes wurde ein graphisches Werkzeug entwickelt und implementiert, welches eine Benutzer-geführte Spezifikation von Transformationsregeln ermöglicht. Dieses wurde um ein methodisches Vorgehen erweitert, welches eine Definition von Transformationsregeln anhand (einfacher), vorgegebener Beispiele beinhaltet und deren Wiederverwendbarkeit in beliebigen Systemkontexten sicherstellt. Hierzu wird der Benutzer durch das Werkzeug schrittweise durch alle erforderlichen Entscheidungen geführt. Abbildung 40 zeigt einen Screenshot verschiedener Dialogfenster der graphischen Benutzeroberfläche des Werkzeugs.



**Abbildung 40: Benutzer-geführte Spezifikation von Transformationsregeln**

Aufgrund der hohen Sicherheitsanforderungen im Automobilbereich ist neben der Automatisierbarkeit auch die Sicherstellung der Richtigkeit von Transformationsregeln eine zen-

trale Fragestellung. Hierzu wurde ein methodisches Vorgehen entwickelt und implementiert, welches es ermöglicht, die Korrektheit („die Transformation macht das richtige“) und Vollständigkeit („die Transformation berücksichtigt alle möglichen Fälle“) von Transformationsregeln sicherzustellen.

Der entwickelte Transformationsansatz wurde zur Realisierung von Lösungsverfahren für eine Reihe von typischen Problemstellungen beim Entwurf automobilelektronischer Netzwerke erfolgreich angewendet. Diese beinhalten zum einen Transformationsregeln zur zielgerichteten Verbesserung der Integrierbarkeit von Systemmodulen in eine vernetzte Systemumgebung (etwa Umsetzung firmenspezifischer Codier-Richtlinien bei der Integration von IP, Generierung von Wrappern zur Schnittstellenanpassung, etc.) und adressieren somit die Subsystem-Ebene. Zum anderen wurden Transformationsregeln betrachtet, die eine zielgerichtete Verbesserung komplexer Systemeigenschaften (wie etwa Zuverlässigkeits-, Sicherheits- oder Fehlertoleranzeigenschaften) auf Ebene des Netzwerks, also auf Gesamtsystem-Ebene, ermöglichen (Schnittstelle zu Beitrag 2.2.1). Als Vertreter dieser zweiten Kategorie wurde eine Transformationsregel realisiert, welche eine Identifikation von Busstrukturen innerhalb eines Netzwerkverbunds vornimmt und diese automatisiert um Fehlererkennungs- und -korrekturmechanismen erweitert. Abbildung 41 zeigt einen Überblick über das Vorgehen, dabei bezeichnen E und D Encodier- bzw. Decodier-Einheiten, die durch die Transformation in Kombination mit einer automatischen Vergrößerung der Busbreite in das Netzwerk integriert werden.

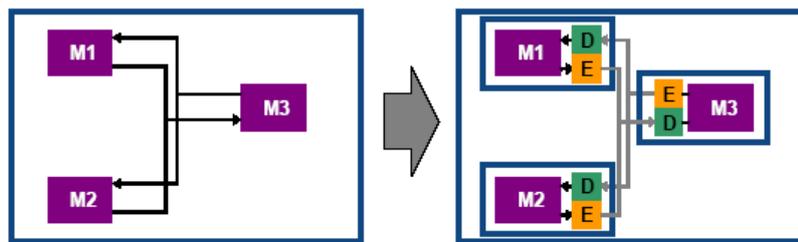


Abbildung 41: Transformation zur Erweiterung von Busstrukturen um Fehlererkennungs- und -korrekturmechanismen

Mittels der realisierten Transformationsregeln konnte die entwickelte Methodik validiert und der hohe Nutzen des Vorgehens nachgewiesen werden.

#### 4.4 AP4: Applikationen und Designflow-Integration

Die in den Arbeitspaketen 1 bis 3 entwickelten Methoden adressierten domänenübergreifende Probleme beim Entwurf verteilter mikroelektronischer Systeme. Die Arbeiten in Arbeitspaket 4 integrierten diese Methoden in einen durchgängigen Entwurfsablauf (Designflow), der die Applizierung der Methoden ermöglichte. Existierende System-Level-Designflows waren nur auf den Entwurf eines einzelnen Systems ausgerichtet und erlaubten nicht, besondere Eigenschaften von verteilten Systemen zu integrieren. Daher wurden in diesem Arbeitspaket, aufbauend auf den Methoden der Arbeitspakete 1 bis 3, applikationsspezifische Designflows für verteilte mikroelektronische Systeme bereitgestellt, welche in der Lage sind, die spezifischen Eigenschaften der adressierten Anwendung und Entwurfsdomäne zu berücksichtigen.

In Arbeitspaket 4 erfolgte die Ableitung von applikationsspezifisch angepassten Designflows aus den allgemeinen Methoden der Arbeitspakete 1 bis 3 (Aufgabe 4.1).

Während bisherige Lösungen für den Systementwurf die besonderen Aspekte verteilter Systeme nicht berücksichtigten (siehe Arbeitspaket 1 bis 3), wurden in Arbeitspaket 4 erstmalig applikationsspezifische Flows für verteilte Systeme erarbeitet. Die industrielle Tauglichkeit wurde dabei durch einen speziellen Schritt zur Anpassung an die jeweilige Applikationsdo-

mäne erreicht. Durch die Bereitstellung exemplarischer Flows konnten zusätzliche Anforderungen für weitere Applikationsdomänen und für die Anbindung anderer Flowkomponenten konkretisiert werden. Diese Anforderungen können an EDA-Toolhersteller weitergegeben werden, sodass hier eine Manifestierung der Methoden und eine Anpassung der EDA-Landschaft zu erwarten ist.

#### **Aufgabe 4.1: Applikationsspezifische Anpassung der Entwurfsmethodik**

##### **Beitrag 4.1.1: Applikationsspezifische Anpassung der entwickelten Entwurfsmethodik an die spezifischen Anforderungen der Applikationsdomäne Automobil**

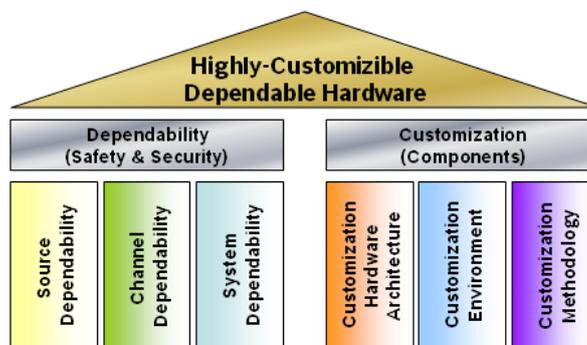
In Beitrag 4.1.1 erfolgte von Bosch eine applikationsspezifische Anpassung der entwickelten Entwurfsmethoden an die spezifischen Anforderungen und Randbedingungen der von Bosch adressierten Applikationsdomäne Automobil. Hierzu erfolgte eine zeitnahe Anwendung der entwickelten Projektmethoden auf ausgewählte Applikationen, welche die spezifischen Anforderungen und Randbedingungen der Applikationsdomäne Automobil in hohem Maße wieder spiegeln. Auf diese Weise flossen die Erprobungsergebnisse aus Beitrag 4.1.1 kontinuierlich in die Methodenentwicklung der anderen Arbeitspakete ein und unterstützten so eine zielgerichtete Weiterentwicklung und Optimierung der Verfahren.

Für die im Rahmen des Aufgabenpakets zu berücksichtigende applikationsspezifische Anforderung wurde die Systemeigenschaft „Systemsicherheit“ in den Teilaspekten Safety und Security ausgewählt. Safety-Mechanismen in automobilelektronischen Systemen adressieren die zielgerichtete Erhöhung der Robustheit des Netzwerks gegenüber Fehlern. Dies kann etwa durch Realisierung von Mechanismen zur Gewährleistung der Sicherheit und Verfügbarkeit kritischer Systemfunktionen im Fehlerfall (beispielsweise durch redundante Auslegung der kritischen Funktionalität) oder durch Realisierung von Mechanismen zur Gewährleistung einer erhöhten Verfügbarkeit bei kontrollierter Einschränkung des Funktionsumfangs im Fehlerfall (beispielsweise durch Deaktivierung von Komfortfunktionen zur Sicherstellung des fehlerfreien Betriebs der kritischen Funktionalität) erreicht werden. Aufgrund des hohen Sicherheits- und Zuverlässigkeitsanspruchs automobilelektronischer Systeme stellt dieser Aspekt eine wichtige Anforderung im Automobilbereich dar. Security-Mechanismen in automobilelektronischen Systemen adressieren den Schutz vor Manipulation, Diebstahl und unbefugter Benutzung / unbefugtem Zugang sowie den Plagiatschutz und Schutz vor nicht-bestimmungsgemäßem Gebrauch. Dieser Aspekt stellt vor dem Hintergrund wirtschaftlicher Schäden sowie Schadensersatzansprüche aufgrund minderwertiger oder unsachgemäß integrierter Systemkomponenten ebenfalls eine wichtige Anforderung in heutigen Automobilnetzwerken dar.

Im Verlauf der Arbeiten in Beitrag 4.1.1 erfolgte eine Umsetzung verschiedener Safety- und Security-Maßnahmen unter Anwendung der entwickelten Methoden. Hierzu wurde eine Hardwareplattform für Safety-/Security-Anwendungen konzipiert, welche die in Aufgabe A2.2 entwickelte Gateway-Architektur zugrunde legt und den Einsatz einer spezialisierten Safety-/Security-Unit vorsieht. Als eine Security-Maßnahme wurde ein Verfahren entworfen, welches die kombinierte Anwendung eines symmetrischen und asymmetrischen Verschlüsselungsverfahrens vorsieht (Authentifizierung der Kommunikationsteilnehmer mittels asymmetrischer Verschlüsselung, anschließende Datenübertragung mittels symmetrischer Verschlüsselung).

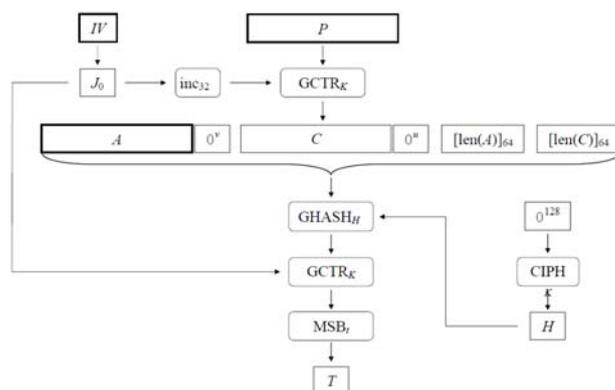
Ausgehend von dem entwickelten Konzept einer generischen Safety/Security-Architektur erfolgte eine Systematisierung des Spezifikations- und Entwurfsvorgangs. Hierzu wurde ein Schichten-/Säulenmodell für den Entwurf von verlässlichen (dependable; Dependability als Kombination von Safety und Security) Architektur definiert (Abbildung 42) und geeignete Maßnahmen für Safety und Security identifiziert. Exemplarisch wurden einzelne Maßnahmen

zur Implementierung von Fehlertoleranzmechanismen, etwa die preemptive Detektion von Hardware-Fehlern oder die Detektion von Hardware-Fehlern während des Betriebs, sowie Maßnahmen zur Implementierung von Security-Diensten, etwa im Hinblick auf die Verwendung geeigneter kryptographischer Primitive oder die Unterstützung unterschiedlicher Angreifer-Modelle, Angreiferszenarien und Schutzklassen innerhalb des systematisierten Vorgehens umgesetzt.



**Abbildung 42: Schichten-/Säulenmodell für den Entwurf verlässlicher Kommunikationsarchitekturen**

Im weiteren Verlauf der Arbeiten erfolgte eine Fokussierung auf Safety-/Security-Mechanismen zur Abwehr von physikalischen Angriffen auf die Implementierung kryptographischer Algorithmen vor dem Hintergrund der Geheimhaltung und/oder Authentifizierung von Nachrichten in komplexen vernetzten Kommunikationsszenarien. In diesem Kontext wurde eine Variante des GMAC-Betriebsmodus des AES-128 Verschlüsselungsverfahrens (FIPS-197) gemäß NIST-Empfehlung NIST SP 800-38D als kryptographisches Verfahren zur Geheimhaltung und/oder Authentifizierung von Nachrichten konzipiert. Unter Anwendung der entwickelten Methoden erfolgte der Aufbau und die Analyse eines virtuellen Prototyps für eine Komponente zur Realisierung des AES Galois Counter Mode (GCM) Verfahrens, Abbildung 43 zeigt einen Systemüberblick. Innerhalb des AES-Algorithmus wurden dabei Maßnahmen gegen Fault Analysis Angriffe integriert. Hierzu wird im GCM Modus eine Nachricht mit Authentifizierungsmechanismen versehen, welche auch in der Lage sind, Safety-relevante Anforderungen (das Erkennen zufälliger Fehler) abzudecken.



**Abbildung 43: Galois Counter Mode Einheit des AES-128 Verschlüsselungsverfahrens**

In Summe beinhalteten die Arbeiten in Beitrag 4.1.1 damit eine umfassende Anwendung der entwickelten Methoden auf eine konkrete und für den Bereich der Automobilelektronik relevante Fragestellung. Die Arbeiten lieferten damit zum einen Erkenntnisse hinsichtlich der Leistungsfähigkeit und Anwendbarkeit der entwickelten Projektmethoden sowie konkrete Hinweise für deren Optimierung und Erweiterung, zum anderen lieferten die Arbeiten wichtige Erkenntnisse hinsichtlich des betrachteten Anwendungsfalls, dem Entwurf verlässlicher Kommunikationsnetzwerke.

### Beitrag 4.1.2: Anpassung der entwickelten Methoden an die Applikationsdomäne „mobile Endgeräte“ sowie Integration derselben zu einem exemplarischen Entwurfsablauf.

Das Arbeitspaket vier des VISION Projektes diente zur projektbegleitenden Vorbereitung der Verwertung. Dementsprechend wurden von Infineon im Beitrag 4.1.2 die entwickelten Methodiken ständig mit Infineon-Anforderungen abgeglichen. Als Zielapplikationsdomäne diente hierbei zunächst der Anwendungsbereich "mobile Endgeräte", jedoch ist die Anpassung der Methodiken an andere Geschäftsbereiche von vorneherein vorgesehen gewesen.

Die Verwertung der VISION Methodiken manifestiert sich in der Implementierung der Methodiken in der Form von Subflows, die dann in den existierenden System-Level Entwurfsflow integriert werden, um diesen zu erweitern. Dementsprechend wurden im ersten Schritt Anforderungen für den erweiterten Flow definiert, welche in die Entwicklung der Methodiken in den Arbeitspaketen eins bis drei eingingen. Diese Flowanforderungen betreffen sowohl die Methodiken direkt (da ja bestimmte Schritte im Entwurfsablauf damit realisiert bzw. verbessert werden sollen) als auch die generelle Infrastruktur, mit der die entwickelten Werkzeuge zusammenspielen müssen.

Im zweiten Schritt wurden dann die mittlerweile entstandenen Methodiken mit Anforderungen der Infineon Geschäftsbereiche abgeglichen und daraus entsprechende Teilflows konzipiert, die dann im dritten Schritt prototypisch implementiert wurden. Die Teilflows, die hierbei als Projektergebnisse erstellt wurden, sind:

- Teilflow für die generelle Codegenerierung aus Beschreibungen der VISION Beschreibungsmethodik mit der Möglichkeit der Einbindung von Fremdformaten (vgl. Beitrag 1.1.3)
- Teilflow zur frühen Performanzanalyse basierend auf EQN\*-Modellen, die von abstrakten Systembeschreibungen abgeleitet werden (vgl. Beitrag 2.1.3)
- Teilflow zur Erstellung von Komponenten für virtuelle Prototypen nach dem "Layered Modelling Approach" (vgl. Beitrag 3.1.2)

Neben der gezielten Konzipierung von Teilflows wurden die Methodiken während der gesamten Projektlaufzeit auf Detailebene an Infineon-Anforderungen angepasst und entsprechend erweitert. Dies trat besonders bei der Beschreibungsmethodik (vgl. Beitrag 1.1.3) zutage, wo aktuelle Entwicklungen gleich in dem VISION-Datenmodell berücksichtigt werden konnten. Die Verzahnung der technischen Arbeiten mit dem Beitrag 4.1.2 ist in Abbildung 44 illustriert.

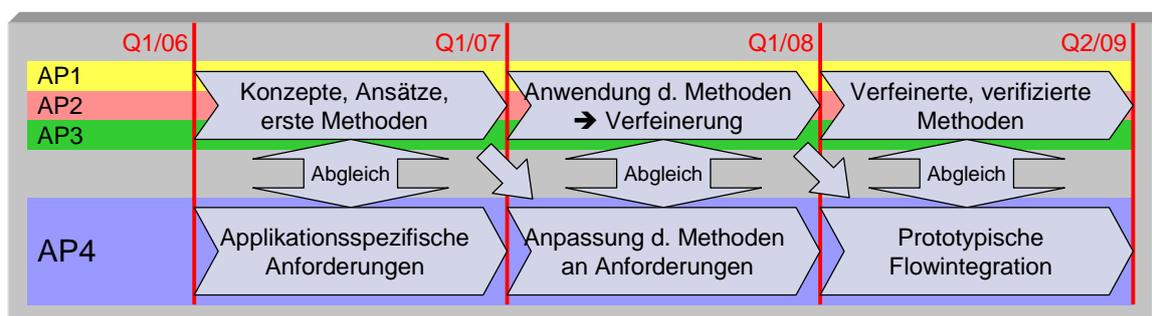


Abbildung 44: Applikationsspezifische Anpassung in VISION

Der Wert der projektbegleitenden Anpassung der Methodiken liegt in der wesentlich besseren Vorbereitung der Verwertung, als wenn diese nach Projektende vorgenommen worden wäre. Insbesondere in einem volatilen Umfeld wie im Falle der Applikationsdomäne "mobile Endgeräte" hat der Beitrag 4.1.2 allein durch das zeitnahe Einbringen von realen Anforderungen während der Projektlaufzeit einige Aufwände nach Projektende für die Verwertungs-

vorbereitung erspart. Beispielsweise hat sich herausgestellt, dass für die Akzeptanz der Generierungsmethodik auch die Einbindung von Fremdformaten notwendig ist (vgl. oben und Beitrag 1.1.3). Durch die frühzeitige Berücksichtigung dieser Anforderung konnte sowohl die Methodik selbst wesentlich flexibler und leistungsfähiger gestaltet werden, als auch die Vorbereitung der Verwertung wesentlich vereinfacht werden.

## **4.5 Voraussichtlicher Nutzen**

### **Robert Bosch GmbH**

#### **Nutzen und Verwertbarkeit Beitrag 1.2.1**

Die als zentrales Ergebnis entstandene Modellierungs- und Analyseplattform kann – wie im Rahmen der Projektarbeiten bereits am Beispiel von FlexRay-System szenarien gezeigt – unmittelbar in einer frühen Phase der Systementwicklung eingesetzt werden. Der Ansatz liefert damit ein wertvolles Hilfsmittel zur Beantwortung zentraler Fragestellungen bei der Architekturfindung und -optimierung. Der Ansatz ermöglicht es, wichtige und weitreichende Architekturentscheidungen auf hoher Abstraktionsebene zu untersuchen und in das abstrakte Modell zurückzuführen. Auf diese Weise kann die gesamte Phase der Architektur-exploration effizient auf hoher Abstraktionsebene durchgeführt werden. Dies führt zu einer deutlichen Reduzierung von Entwicklungszeit und Entwicklungskosten im Vergleich zur bisherigen Vorgehensweisen. Durch die Bereitstellung von Schnittstellen zu Matlab/Simulink und SystemC werden dabei wichtige der im industriellen Umfeld eingesetzten System-beschreibungsansätze sowie das, sich zwischenzeitlich zum Standard etablierte, Vorgehen zum Austausch und zur Wiederverwendung eigener und fremder Komponentenmodelle unterstützt. Durch die Möglichkeit zur automatischen Generierung ausführbarer SystemC-Modelle stehen Mechanismen zur effizienten Erzeugung von Referenzmodellen für nachfolgende Entwurfsschritte zur Verfügung. Entsprechend einfach gestaltet sich die Integration des Ansatzes in bestehende Entwicklungsabläufe sowie dessen Anbindung an nachfolgende domänenspezifische Implementierungsprozesse. Dies führt zu einer Erhöhung der Entwurfs-sicherheit und Steigerung der Entwurfseffizienz.

#### **Nutzen und Verwertbarkeit Beitrag 2.2.1**

Die entwickelte Architektur eines Kommunikationscontrollers sowie die entstandene Konfigurations- und Testumgebung liefern einen wichtigen Beitrag für einen strukturierten Entwurf von sicheren und zuverlässigen Kommunikationsnetzwerken. Dies führt zu einer Erhöhung der Entwurfssicherheit und daraus resultierend zu einer Verbesserung der Ergebnisqualität. Das entwickelte Controllerkonzept stellt den Ausgangspunkt für weitere Entwicklungsarbeiten dar, deren Endziel es ist, das entwickelte Konzept für einen produktiven Einsatz verfügbar zu machen. Die entwickelte Konfigurations- und Testumgebung stellt ein Rahmenwerk zur Untersuchung aktueller und zukünftiger Verbindungsstrukturen zur Verfügung und liefert so ebenfalls einen wichtigen Beitrag zur Erhöhung der Qualität und Zuverlässigkeit von elektronischen Netzwerken.

#### **Nutzen und Verwertbarkeit Beitrag 3.1.1**

Im Rahmen der Arbeiten ist eine SystemC-basierte Methodik entstanden, die in modularer Weise eine Verifikation komplexer Systemkomponenten sowie deren Integration in ein vernetztes Gesamtsystem unterstützt. Der entwickelte Ansatz ermöglicht somit die Validierung eines auf virtuellen Prototypen basierenden abstrakten Systemmodells – Nutzer dieser Technik sind Systemarchitekten in einer frühen Phase des Systementwurfsprozesses – bis hin zur Verifikation von Hardware-Implementierungen der Systemmodelle auf Register-Transfer-Ebene – Nutzer sind hier die Schaltungsentwickler. Durch Anwendung einer ebenenübergreifenden Verifikationsmethodik kann dabei die Konsistenz von Entwurfs-

schritten gewährleistet und die Durchgängigkeit des Entwurfsprozesses erhöht werden. Damit liefert der Ansatz ein wichtiges Instrument zur Erhöhung der Effizienz und Reduzierung des Fehlerpotenzials an der Schnittstelle zwischen System-/Steuergeräteentwicklung und Halbleiterentwicklung. Weiter bietet der Ansatz eine komfortable Schnittstelle zu benachbarten Entwurfsdomänen (etwa Analog/Mixed-Signal oder Software) und Entwurfsaktivitäten (etwa Test und Integration) und liefert so einen wichtigen Beitrag zum Aufbau eines ganzheitlichen Systemverständnisses. Das entwickelte Vorgehen wird zum einen im Rahmen von Pilotanwendungen in den industriellen Entwurfsablauf von integriert und liefert zum anderen die Grundlage für weiterführender Forschungs- und Entwicklungsarbeiten.

#### **Nutzen und Verwertbarkeit Beitrag 3.2.1**

Das entwickelte ganzheitliche Vorgehen zur zielgerichteten Transformation von Systembeschreibungen zeichnet sich durch einen extrem hohen Grad an Flexibilität aus. Dies eröffnet ein breites Spektrum von Anwendungsmöglichkeiten im Bereich der System- und Hardwareentwicklung, deren Transfer in den industriellen Entwurfsablauf Gegenstand aktueller Arbeiten ist. Das Vorgehen ermöglicht eine Automatisierung von Entwurfsschritten und führt so zu einer Erhöhung der Entwurfsproduktivität. Weiter wurden Strategien erarbeitet, welche den Nachweis der Korrektheit und Vollständigkeit von Entwurfsschritten zum Inhalt haben und so einen wichtigen Beitrag zur Erhöhung der Entwurfssicherheit und damit Steigerung der Sicherheit und Zuverlässigkeit des Entwurfsergebnisses liefern.

#### **Nutzen und Verwertbarkeit Beitrag 4.1.1**

Die im Projekt in Aufgabenpaket A4.1 durchgeführte applikationsspezifische Anpassung der entwickelten Projektmethoden an die spezifischen Anforderungen und Randbedingungen der Applikationsdomäne Automobil ermöglichte eine zeitnahe Erprobung der entwickelten Ansätze anhand konkreter und relevanter Fragestellungen. Auf diese Weise wurde gewährleistet, dass aktuelle Trends und Entwicklungen im Automobilbereich unmittelbar in die Methodenentwicklung in den Arbeitspaketen des Projekts einfließen. Durch diesen engen und fortlaufenden Anwendungsbezug wurde eine spätere Verwertung der Projektergebnisse bestmöglich vorbereitet. Weiter wurden durch die Bearbeitung konkreter Fragestellungen in den jeweiligen Anwendungsbereichen wichtige Erkenntnisse gewonnen, die zukünftig vorteilhaft – unter Verwendung der entwickelten Projektmethoden – für den Aufbau von innovativen Systemlösungen eingesetzt werden können.

### **Cadence Design Systems GmbH**

Die Cadence Design Systems GmbH entwickelt neben dem Verkauf und Support von EDA-Werkzeugen auch eigenständig neue Softwarelösungen. Weiter bietet Cadence Dienstleistungen im Bereich der Schaltungsentwicklung, sowie der Entwicklung von kundenspezifischen Entwurfsumgebungen.

Zum Ende des VISION Projektes steht eine verifizierte Methodik zur Verfügung, deren Einsatz beispielhaft innerhalb der Entwurfsumgebung überprüft wurde. Die Forschungsergebnisse ermöglichen es Cadence Deutschland und insbesondere deren Methodikgruppe, ihr Arbeitsfeld in dem Bereich Systemdesign weiter zu vertiefen. Die Erkenntnisse aus den Arbeiten in Arbeitspaketen 1, 2 und 3 können nunmehr in kundenspezifische Verfahren zur Spezifikation, Modellierung und Simulation bzw. Verifikation einfließen.

#### **Nutzen und Verwertbarkeit Beitrag 1.1.1**

Im Rahmen dieser Arbeiten wurde ausgehend von einer funktionalen Beschreibung eines Systems in SysML/UML eine automatisierte Generierung eines funktionalen C++ sowie eines SystemC Modells einschließlich einer Simulationsumgebung vorgenommen, die in der Architekturfindung und –bewertung eingesetzt werden kann. Durch die automatisierte Generierung von ausführbaren SystemC Modellen aus der abstrakten Systemspezifikation können

bereits zu einem frühen Zeitpunkt Aussagen über das entworfene System getroffen und Spezifikationsverletzungen aufgezeigt werden. Dieser Ansatz erlaubt es, grundlegende Architekturentscheidungen auf einer hohen Abstraktionsebene und somit frühzeitig im Entwicklungsprozess zu treffen. Die Entwicklungszeiten und die damit verbundenen Kosten können deutlich gemindert werden.

### **Nutzen und Verwertbarkeit Beitrag 1.2.2**

In diesem Beitrag wurde bei Cadence eine busähnliche Kommunikationstopologie aufgebaut, die es in einer frühen Phase im Entwicklungsprozess ermöglicht, ein in SystemC implementiertes simulationsfähiges Modell aufzustellen. Es wurden Monitore entwickelt, die die Beobachtung und Überwachung der Kommunikation, d.h. des Zusammenspiels einzelner Komponenten innerhalb eines verteilten Systems übernehmen. Diese Monitore wurden zusätzlich mit verschiedenen Schnittstellen versehen, um sowohl eine ereignisgetriebene wie auch eine zeitgesteuerte Aktivierung in der Simulation zu unterstützen. Im Weiteren wurde die Funktionsweise eines Werkzeuges zur automatischen Generierung von Kommunikationsmonitoren sowie deren Anbindung an eine Simulationsumgebung beschrieben. Somit ist eine automatisierte Beurteilungsinfrastruktur entstanden, mittels derer das zu entwerfende System frühzeitig bezüglich der Spezifikationsanforderung bewertet werden kann.

### **Nutzen und Verwertbarkeit Beitrag 2.1.1**

Der in diesem Beitrag vorgestellte Prozess bietet eine durchgehende simulative Vorgehensweise, wie man von einer in SysML/UML beschriebenen funktionalen Beschreibung eines Systems über verschiedene Verfeinerungsschritte zu einem Architekturmodell findet. Zur Bewertung und Überprüfung der Einhaltung des Anforderungsprofils wird jedes Modell durch Simulation in der gleichen Umgebung und unter Ausnutzung derselben Testsuite (corner test und zufällig generierten Stresstests) überprüft. Die Ergebnisse der Simulationen erlauben eine Bewertung der jeweiligen Architekturalternative sowohl von Software- als auch von Hardware- Komponenten schon zu einem frühen Zeitpunkt im Entwicklungsprozess – langwierige und kostenintensive Iterationsschleifen lassen sich erheblich reduzieren.

### **Nutzen und Verwertbarkeit Beitrag 3.1.2**

Die im Beitrag entwickelte Verifikationsmethode unterstützt das ganzheitliche Vorgehen bei der System- bzw. Hardwareentwicklung. Die aufgezeigte Methode basiert auf wohl definierten und konfigurierbaren Verifikationskomponenten, die hierarchisch zusammengefügt und auf unterschiedlichen Abstraktionsebenen eingesetzt werden können. Somit wird ein hohes Maß an Wiederverwendbarkeit der einzelnen Komponenten während der unterschiedlichen Phasen des Designprozesses erzielt. Die Ebenenunabhängigkeit erlaubt es schon frühzeitig im Entwicklungsprozess eine Verifikation der Systemeigenschaften vorzunehmen. Die vorgestellte Methode kann projektübergreifend eingesetzt werden und führt zu einer Erhöhung der Entwurfssicherheit und Steigerung der Entwurfs-effizienz.

## **FZI Forschungszentrum Informatik**

Die Ergebnisse des Projekts VISION haben das FZI in die Lage versetzt, den Entwurfsablauf komplexer vernetzter Systeme durch Erforschung und Entwicklung einer ganzheitlichen Modellierungsumgebung für verteilte mikroelektronische Systeme stark zu verbessern, welcher die Verfeinerung des Systems, ausgehend von der Modellebene bis hin zum virtuellen Prototypen, mit einer umfassenden Bewertung von Architektur und Netzwerk auf unterschiedlichen Abstraktionsebenen unterstützt. Allein durch die Tatsache, dass viele Integrationsfehler ohne die aufwändige Umsetzung eines realen Prototypen aufgedeckt werden können und der Entwurfsaufwand für eine implementierbare Spezifikation mindestens um eine Größenordnung höher ist als für ein abstraktes Modell, kann der Entwurf komplexer vernetzter, mikroelektronische Systeme beherrschbar gemacht werden. So konnte bereits während der

Projektlaufzeit, durch Einsatz der neu erforschten Methoden und Werkzeuge, mehrere komplexe Designs aus dem Automobilumfeld erfolgreich abgeschlossen werden, wie z. B. das Demonstratorsystem zur Verkehrszeichenerkennung inklusive der Algorithmenentwicklung, der Integration der AUTOSAR-konformen Software-Komponenten sowie der Beschreibung der Hardware- und Netzwerk-Architektur unter Verwendung von Modellen der Plattformkomponenten.

Aus wissenschaftlicher Sicht konnte das FZI seine starke internationale Rolle im Bereich ESL-Entwurfsmethoden festigen und ausbauen und den Standardisierungsprozess des TLM2.0-Standards aktiv durch Mitarbeit in der European SystemC User Group sowie über GreenSocs mitgestalten.

### **Nutzen und Verwertbarkeit Beitrag 1.1.2**

Der im Beitrag entwickelte Ansatz erlaubt eine ganzheitliche Beschreibung des Gesamtsystems, einschließlich der Modellierung der Zielplattform, der Software, der Kommunikation, sowie des Zeitverhaltens, auf unterschiedlichen Abstraktionsebenen und unterstützt somit eine spezifikationsgetriebene Modellierung verteilter eingebetteter Systeme, welche zuvor nur separiert für einzelne Systemaspekte wie Hardware oder Software möglich war. Dadurch ist es bereits zu einem sehr frühen Zeitpunkt möglich, Integrationsfehler zu erkennen und dadurch Entwurfsiterationen und Entwicklungskosten zu reduzieren. Der Ansatz ermöglicht weiterhin eine enge Verzahnung des Entwurfs der verteilten Hardwarearchitektur sowohl mit dem Requirements-Engineering als auch mit dem Entwurf der eingebetteten Software und unterstützt somit den nahtlosen Übergang von der Spezifikationsphase zu den Implementierungsprozessen. Dies verspricht eine Beschleunigung des Entwurfs und eine reduzierte Time-to-Market. Zudem unterstützt der Ansatz eine strukturierte und konstruktive Wiederverwendung von Komponenten (IP). Durch Ausrichtung des Ansatzes an etablierte Modellierungsstandards wie UML, XML, MARTE und IP-XACT kann der Ansatz schnell und einfach in etablierte Entwurfsabläufe weiterer Applikationsdomänen integriert, sowie bei anderen Firmen begleitend eingeführt werden. Durch die automatisierte Generierung von ausführbaren SystemC-Modellen aus der abstrakten Systemspezifikation können bereits zu einem frühen Zeitpunkt Aussagen über das entworfene System getroffen werden und Spezifikationsverletzungen aufgezeigt werden.

### **Nutzen und Verwertbarkeit Beitrag 1.2.3**

Die im Rahmen dieses Beitrags entwickelten Methoden und Werkzeuge erlauben eine Template-basierte Verfeinerung von Kommunikationen auf Busstrukturen, sowie die Generierung eines dem verfeinerten Modell entsprechenden virtuellen Prototypen zur simulationsbasierten Bewertung der unterschiedlichen Bus-Zugriffsverfahren. Die Verfahren und Werkzeuge können in frühen Phasen des Entwurfs eingesetzt werden. Sie beschleunigen die Verfeinerung von Entwürfen durch eine automatisierte Umsetzung von Entwurfsentscheidungen auf Modellebene, sowie auf der Ebene virtueller Prototypen. Daraus ergibt sich eine signifikante Reduktion der Implementierungskosten zur Prüfung der Auswirkung einzelner Entwurfsentscheidungen auf das globale Systemverhalten, sowie eine damit verbundene deutliche Reduktion der Entwurfszeit und der Entwurfskosten. Durch Einbindung und Wiederverwendung von Komponenten (IP), welche der getroffenen Entwurfsentscheidung entsprechen, wird zudem eine Reduktion der Implementierungszeit unterstützt.

Die im Beitrag entwickelte Methodik zur semiautomatischen Topologiefindung, -generierung, -verfeinerung und -optimierung unterstützt den Entwickler bei der Auswahl einer maßgeschneiderten Kommunikationsinfrastruktur für das verteilte eingebettete System. Die analytische Vorgehensweise unterstützt die Reduktion der Anzahl manueller Entwurfsiterationen, beschleunigt somit den Entwurfsprozess und erlaubt ein Vorziehen der nachfolgenden Entwurfsschritte, wodurch eine kürzere Time-to-Market erreicht werden kann. Durch die entwickelten Strategien zur Topologieexploration ist nun auch die Möglichkeit gegeben, die Hardwarearchitektur unter Berücksichtigung der darauf ausgeführten Software zu evaluieren und

dadurch die Zielplattform zu optimieren, was weitere Ressourcen und Kosten einspart, beziehungsweise bisher gar nicht angewandt wurde, da ein durch die Optimierung notwendiges Re-Design der bereits implementierten Zielplattform mit enormem Mehraufwand verbunden gewesen ist. Die kontinuierlich wachsende Anzahl von kommunizierenden Komponenten in verteilten eingebetteten Systemen verspricht zukünftig eine hohe Nachfrage an Methoden, die den Entwickler bei Entwurfsentscheidungen unterstützen und Anpassungshilfen bereit stellen. Hierfür stellt der entwickelte Ansatz eine erste Grundlage zur Verfügung.

### **Nutzen und Verwertbarkeit Beitrag 2.1.2**

Die im Rahmen des Beitrags erforschten Methoden ermöglichen eine frühe Echtzeit- und Leistungsfähigkeitsanalyse verteilter eingebetteter Systeme unter Berücksichtigung der Eigenschaften der Verbindungsarchitektur, der eingesetzten Kommunikationsprotokolle, sowie des eingesetzten Softwareschedulings auf Verarbeitungseinheiten. Basierend auf der Vorgehensweise ist es möglich, ein ganzheitlich spezifiziertes Modell eines eingebetteten mikroelektronischen Systems auf Performanzgrößen wie Auslastung oder Ende-zu-Ende-Latenzen zu prüfen und harte Echtzeitanforderungen zu verifizieren. Durch die Verwendung desselben Spezifikationsmodells, aus dem auch die funktionale Entwicklung des Systems startet, wird eine entwicklungsbegleitende Verifikation von Echtzeiteigenschaften unterstützt, die in jedem Iterationsschritt eine Anforderungsbewertung ermöglicht. Daraus können bereits zu sehr frühen, jedoch auch zu späteren Entwurfszeitpunkten Anforderungsverletzungen identifiziert und längere Iterationen vermieden werden.

Durch das Zusammenbringen funktionaler Komponenten und der modellbasiert spezifizierten Hardwareplattform weist die entwickelte formale-analytische Vorgehensweise eine deutliche Beschleunigung der Echtzeitverifikation gegenüber einer simulationsbasierten Vorgehensweise auf, so dass eine Verkürzung der Entwurfszeit und damit einhergehend eine Reduktion der Entwurfskosten erreicht werden kann. Zudem können Garantien über das Einhalten von Grenzfällen gegeben werden, welche essentiell für harte Echtzeitanforderungen sind. Das entwickelte Vorgehen kann in industriellen Pilotprojekten angewendet werden und liefert Grundlagen für weiterführende Forschungs- und Entwicklungsarbeiten.

### **Infineon Technologies AG**

Infineon wird die Projektergebnisse von VISION schrittweise in den qualifizierten Designflow integrieren und damit die Produktivität beim Entwurf komplexer Systemverbunde verbessern. Als Treiber und Beispielapplikation für das Projekt wurden moderne Mobilfunk-Terminals gewählt, darüber hinaus werden die Ergebnisse nach der globalen Einführung in den Flow jedoch auch bei anderen Geschäftsfeldern zum Einsatz kommen.

Um die Verwertung für den Bereich der Mobiltelefonie vorzubereiten, arbeiteten in VISION Experten der Methodikgruppen für System Level mit Experten des Geschäftsbereiches „Communications“ zusammen, um zum einen den Applikationsbezug und zum anderen die zentrale Verfügbarkeit der Methoden zu gewährleisten.

Mit den Ergebnissen von VISION begegnet Infineon der Verschärfung des Wettbewerbs im Bereich der Mobilkommunikation, bei dem der Erfolg durch die Faktoren Entwurfszeit, Funktionalität, Entwurfsqualität und Kundenzufriedenheit beeinflusst wird.

Die Projektergebnisse von VISION adressieren diese Faktoren, was im Folgenden kurz skizziert werden soll:

### **Nutzen und Verwertbarkeit Beiträge 1.1.3 und 1.2.4**

Die Beschreibungsmethodik und die daran angeschlossene Generierungsmethodik erhöhen die Entwurfsqualität durch die formale Darstellung von Eigenschaften, da hierbei Missverständnisse und Mehrdeutigkeiten vermieden werden, welche zu Fehlern führen können. Ins-

---

besondere die Generierungsmethodik vermeidet fehleranfälliges und mühsames manuelles Abtippen von Daten. Damit wird sowohl wiederum die Entwurfsqualität als auch die Entwurfszeit verbessert. Dieser Effekt wird besonders deutlich, wenn man das "Nachziehen" von Änderungen an den Entwurfsdaten betrachtet. Muss dies manuell durchgeführt werden, so ist dies mit einem hohen Aufwand von typischerweise mehreren Personen verbunden, wobei die Gefahr für Inkonsistenzen bei den einzelnen Prozessschritten sehr hoch ist. Durch die automatische Generierung von Entwurfsdaten aus einer einzigen Beschreibung wird dies wesentlich vereinfacht. Durch die Verbesserung der Entwurfsqualität ergibt sich natürlich auch direkt eine höhere Kundenzufriedenheit.

### **Nutzen und Verwertbarkeit Beitrag 2.1.3**

Die Methodik zur Performanzanalyse erlaubt die frühe Untersuchung von zusätzlicher Funktionalität und die Auswirkung auf das Gesamtsystem. Hierdurch kann die durch die funktionelle Erweiterung der Systemkomplexität gehandhabt werden. Weiterhin werden durch die frühe Analyse auch Fehler früher erkannt und verursachen dadurch weniger Aufwand bei ihrer Korrektur, was sowohl die Entwurfsqualität als auch die Entwurfszeit verbessert.

### **Nutzen und Verwertbarkeit Beitrag 3.1.3**

Die Methodik für Komponenten für virtuelle Prototypen erlaubt die effizientere Erstellung von solchen Komponenten. Da hierbei Teile der Komponente wiederverwendet werden, brauchen diese nicht für jede Abstraktionsebene neu entwickelt zu werden. Damit entfällt sowohl der Implementierungs- als auch der Verifikationsaufwand (Entwurfszeit und Entwurfsqualität). Weiterhin kann der virtuelle Prototyp an Kunden ausgeliefert werden, welche bereits vor der Verfügbarkeit des Siliziums die Software entwickeln können, was die Time-to-Market dieser Kunden verkürzt und ihre Zufriedenheit erhöht. Weiterhin werden virtuelle Prototypen auch intern für die Software-Entwicklung und als Referenz für die Hardware-Entwicklung verwendet, sodass wiederum die Entwurfszeit verkürzt und die Entwurfsqualität erhöht wird.

### **Nutzen und Verwertbarkeit Beitrag 4.1.2**

Wie bereits in der technischen Beschreibung dargestellt, wurde durch den kontinuierlichen Abgleich der entwickelten Methodiken mit Anforderungen der Geschäftsbereiche die Anpassungsphase nach Projektende verkürzt, sodass die Projektergebnisse schneller der Verwertung zugeführt werden konnten. Durch diesen Abgleich ergaben sich auch erst die Konzepte für die drei oben dargestellten Subflows, in die die Einzelbeiträge zusammenfließen. Zusammenfassend stellt dieser Beitrag somit die Kanalisierung der technischen Beiträge in verwertbare Projektergebnisse in der Form prototypischer Subflows dar.

Tabelle 2 zeigt im Vergleich zu Tabelle 1, wie die zu Beginn des Projektes definierten Ziele bearbeitet wurden und welches technische Ergebnis dabei erreicht wurde.

Problemfelder	Entwurfsprozess	Erzielte Ergebnisse
Modellierung verteilter Systeme	Keine vollständige Modellierung verteilter Systeme möglich, sondern unabhängige Modellierung der Teilsysteme	<ul style="list-style-type: none"> <li>Exploration der Netzwerktopologie von FlexRay-Topologien mit optimierter, konfliktfreier Device-Konfiguration</li> <li>Automatische Exploration der Netzwerktopologie</li> </ul>
Festlegung der Netzwerkstruktur	Manuelle Bestimmung der Verbindungstopologie und manuelle Festlegung der Freiheitsgrade	<ul style="list-style-type: none"> <li>Netzwerkanalyse unter Berücksichtigung des Protokollverhaltens</li> <li>Frühe Performanzanalyse basierend auf abstrakten Systembeschreibungen</li> </ul>
Bewertung des Gesamtsystemverhaltens	Bewertung des Gesamtsystems erst mit realen Prototypen möglich	<ul style="list-style-type: none"> <li>Effiziente VP Erstellung durch automatische Generierung und Reuse</li> <li>Verifikation von Netzwerkkomponenten und deren Integration in ein vernetztes System auf Basis abstrakter virtueller Prototypen</li> </ul>
Verifikation der Systemintegration	Verifikation des Gesamtsystems erst am realen Prototypen möglich	<ul style="list-style-type: none"> <li>Flexible Generierungsmethodik für alle Implementierungsschritte</li> <li>Transformation von Systemmodellen zur Abbildung komplexer System- und Netzwerkeigenschaften in den Systementwurfsablauf</li> </ul>
Einbettung in den Implementierungsprozess	Keine durchgängige Implementierungsprozesse für verteilte Systemlösungen verfügbar	<ul style="list-style-type: none"> <li>Flexible Generierungsmethodik für alle Implementierungsschritte</li> </ul>

Tabelle 2: Erzielte Ergebnisse von VISION hinsichtlich der Entwurfbarkeit

#### 4.6 Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen

Neue im Projektverlauf bekannt gewordenen Erkenntnisse und Lösungsansätze aus dem kommerziellen und akademischen Bereich wurden kontinuierlich in die Projektarbeiten eingearbeitet und finden sich auf diese Weise in den heutigen Methoden wieder. Dies betrifft insbesondere die Weiterentwicklung von SystemC (etwa hinsichtlich TLM 2.0) und SystemC-basierten Werkzeugen. Während der Projektlaufzeit wurden keine Fortschritte an anderen Stellen bekannt, die zu einer Beendigung oder wesentlichen Anpassung der geplanten Aktivitäten geführt hätten.

---

## 5 Veröffentlichungen und Patente

Im Projekt erzielte Ergebnisse wurden auf nationalen und internationalen Konferenzen und Workshops präsentiert. Die insgesamt 51 Veröffentlichungen untergliedern sich in

- 1 Pressemitteilung
- 6 Beiträge zu Fachmagazinen
- 44 Beiträge auf Konferenzen und Workshops

diese in 8 Ländern (darunter USA, Japan, Vereinigte Arabische Emirate, Kanada)  
davon 1 eigene Session auf der FDL08  
davon 1 Workshop auf der DAC09

- (1) T. Lorenz, J. Taube, M. Ihle, O. Manck, H. Beikirch, "FIBEX Gateway Configuration Tool Chain". 11th international CAN Conference, September 26-28, 2006, Stockholm (Sweden), Proceedings p. 04-13...04-19
- (2) O. Bringmann, J. Gerlach, U. Nageldinger, F. Schäfer, "Überblick über das Ekompas-Projekt VISION", URANOS Workshop, Hannover, 26. Oktober 2006
- (3) O. Bringmann, J. Gerlach, U. Nageldinger, F. Schäfer, "System Planning Aspekte aus Sicht des Projekts VISION", Kooperationsworkshop System Planning, Hannover, 30. November 2006
- (4) O. Bringmann, "Application-Specific Design of Reliable NoC Architectures with Hard Time Constraints, Panel Network on Chip at 15th IP-Based SoC Design Conference and Exhibition (IP-SOC), Grenoble, 2006.
- (5) O. Bringmann, "Applikationsspezifische Modellierung und Bewertung von Netzwerken verteilter Steuergeräte", GMM-Automotive Forum, Electronica, 15.11.2006, München
- (6) O. Bringmann, J. Gerlach, U. Nageldinger, F. Schäfer, D. Treytnar, "Systeme zuverlässiger vernetzen", Elektronik 26/2006, Seite 23, 22. Dezember 2006
- (7) O. Bringmann, J. Gerlach, U. Nageldinger, F. Schäfer, D. Treytnar, "VISION – Verteilte integrierte Systeme und Netzwerkarchitekturen für die Applikationsdomänen Automobil und Mobilkommunikation", newsletter edacentrum 04 2006, Seite 10-11, Januar 2007
- (8) T. Lorenz, J. Taube, M. Ihle, O. Manck, H. Beikirch, "Verification Environment for Automotive Gateways". Proceedings embedded world Conference 2007, WEKA Verlag Poing, 13.-15- Februar 2007
- (9) O. Bringmann, A. Viehl: "SystemC-Based Architectural Exploration" (poster and software demonstration), Design Automation and Test in Europe (DATE), April 16-20, 2007, Nice, France.
- (10) O. Bringmann, "Automated Generation of the GreenBus TLM Bus Fabric from IP-XACT Specifications", 4th Open Bi Annual Meeting of the GreenSocs Open source initiative GreenSocs, Fringe Meeting, Design Automation and Test in Europe (DATE), 16-20 April 2007, Nizza, Frankreich
- (11) W. Klingauf, R. Günzel, O. Bringmann, P. Parfuntseu, M. Burton, "GreenBus - A Generic Interconnect Fabric for Transaction Level Modelling", Design Automation Conference (DAC), Juli 2006, San Francisco, CA, USA
- (12) M. Krause, O. Bringmann, A. Hergenhan, G. Tabanoglu, W. Rosenstiel, "Timing Simulation of Interconnected AUTOSAR Software-Components", Design Automation and Test in Europe (DATE), 16-20 April 2007, Nizza, Frankreich

- 
- (13) M. Krause, O. Bringmann, W. Rosenstiel, "Target Software Generation: An Approach for Automatic Mapping of SystemC Specifications onto Real-Time Operating Systems", Design Automation for Embedded Systems (DAES), Volume 10, Issue 4, Springer Verlag, 2007
  - (14) L. Lavagno, C. Passerone, F. Schäfer, "Partitioning" in G. Martin, B. Bailey, und A. Piziali (Editors) "ESL Design and Verification - A Prescription for Electronic System Level Methodology (Systems on Silicon), Elsevier LTD, Oxford, 2007
  - (15) R. Lissel, J. Gerlach: "Introducing New Verification Methods into a Company's Design Flow: An Industrial User's Point of View", Design Automation and Test in Europe (DATE), April 16-20, 2007, Nice, France.
  - (16) R. Lissel, J. Gerlach: "SystemC-Based System Verification" (poster and software demonstration), Design Automation and Test in Europe (DATE), April 16-20, 2007, Nice, France.
  - (17) J. Gerlach, R. Lissel: "Einführung neuer Verifikationsmethoden in den industriellen Entwurfsablauf", edaWorkshop'07, June 19-20, 2007, Hannover, Germany.
  - (18) W. Rosenstiel, "IP-XACT Specification of Generic Interconnects", Panel "GreenSocs: Collaborative Work on ESL Infrastructure, Presentation, User Experience and Ecosystem" at Exhibition Theatre, Design Automation and Test in Europe (DATE), Exhibition Theatre: GreenSoCs, 18. April 2007, Nizza, Frankreich
  - (19) F. Schäfer, O. Bringmann, J. Gerlach, U. Nageldinger, "SystemC Based Design Methods for Distributed Microelectronic Systems in Automotive and Communication Domains", MEDEA+ Design Automation Conference, Mai 2007, Grenoble
  - (20) T. Schönwald, J. Zimmermann, O. Bringmann, W. Rosenstiel, "Ein Framework zur automatisierten Generierung von Network-on-Chips in SystemC", edaWorkshop, Hannover, 19-20 Juni 2007
  - (21) T. Schönwald, J. Zimmermann, O. Bringmann, W. Rosenstiel, "Fully Adaptive Fault Tolerant Routing Algorithm for Network-on-Chip Architectures", Euromicro Conference on Digital System Design (DSD), 2007, Lübeck
  - (22) Siebenborn, A. Viehl, O. Bringmann, W. Rosenstiel, "Control-Flow Aware Communication and Conflict Analysis of Parallel Processes", Asia and South Pacific Design Automation Conference (ASP-DAC), 23-26 Januar 2007, Yokohama, Japan
  - (23) Viehl, O. Bringmann, W. Rosenstiel, "Performance-Aware Communication Architecture Synthesis", Proceedings of 14th Workshop on Synthesis and System Integration of Mixed Information Technologies (SASIMI), Sapporo, 2007
  - (24) T. Schönwald, O. Bringmann, W. Rosenstiel: "Region-Based Routing Algorithm for Network-on-Chip Architectures", Proceedings of 25th IEEE NORCHIP Conference, Aalborg Dänemark, 2007
  - (25) O. Bringmann, J. Gerlach, U. Nageldinger, F. Schäfer, „Überblick über das Projekt VISION“, Poster auf dem edaWorkshop07, 19.-20.6.2007, Hannover
  - (26) Kerstan, N. Bannow, W. Rosenstiel; „Enhanced System Analysis“; FDL, September 2007
  - (27) R. Lissel, J. Gerlach: „Introducing new verification methods into a design flow: an industrial user's view“, EDA Tech Forum, volume 4, issue 3, September, 2007
  - (28) O. Bringmann, J. Gerlach, U. Nageldinger, F. Schäfer, D. Treytnar, "VISION: Beherrschung des Entwurfs zukünftiger elektronischer Systemen für eine vernetzte Welt", newsletter edacentrum 01 2008, Seite 5-10, Januar 2008, Hannover
-

- 
- (29) J. Zimmermann, O. Bringmann, J. Gerlach, F. Schäfer, U. Nageldinger, „Holistic System Modeling and Refinement of Interconnected Microelectronic Systems“, MARTE Workshop auf der DATE, 11.3.2008, München
  - (30) W. Liehr, K. L. Buchenrieder, "Performance Evaluation of HW/SW-System Alternatives", Poster auf der DATE08, 11.-13.3.2008, München
  - (31) J. Zimmermann, O. Bringmann, U. Nageldinger, J. Gerlach, F. Schäfer, W. Rosenstiel, „Modellbasierte Plattformkomposition und Generierung virtueller Prototypen für vernetzte integrierte Systeme“, Poster auf dem edaWorkshop08, 6.-7.5.2008, Hannover
  - (32) Kerstan, N. Bannow, W. Rosenstiel, „Closing the Gap in the Analysis and Visualization of Simulation Data for Automotive Video Applications“, Poster auf dem edaWorkshop08, 6.-7.5.2008, Hannover
  - (33) J. Gerlach „VISION: Beherrschung des Entwurfs zukünftiger elektronischer Systeme für eine vernetzte Welt“, edaWorkshop08, 6.-7.5.2008, Hannover
  - (34) J. Schnerr, O. Bringmann, A. Viehl, W. Rosenstiel: "High-Performance Timing Simulation of Embedded Software", Proceedings of 45th Design Automation Conference (DAC), Anaheim, 2008.
  - (35) Wolfgang Ecker, Volkan Esen, Ulrich Nageldinger, Thomas Steininger, Michael Velten: "UML based Code Generation for the HW/SW Interface"; 5th International UML for SoC Design Workshop (UML-SoC 2008), Anaheim, CA, June 8, 2008.
  - (36) Andreas W. Liehr, Heike S. Rolfs, Klaus J. Buchenrieder, Ulrich Nageldinger: "Generating MARTE Allocation Models from Activity Threads"; Forum on Specification and Design Languages 2008 (FDL08), Stuttgart, 2008.
  - (37) Christian Kerstan, Nico Bannow, Wolfgang Rosenstiel: „Enabling Automated Code Transformation and Variable Tracing“, Forum on Specification and Design Languages 2008 (FDL'08), September 23-25, 2008, Stuttgart, Germany.
  - (38) Kai Hylla, Jan-Hendrik Oetjens, Wolfgang Nebel: "Using SystemC for an Extended MATLAB/Simulink verification flow", Forum on Specification and Design Languages 2008 (FDL'08), September 23-25, 2008, Stuttgart, Germany.
  - (39) Jochen Zimmermann, Oliver Bringmann, Joachim Gerlach, Florian Schäfer, Ulrich Nageldinger: "Comprehensive Platform and Component Modeling of Heterogeneous Interconnected Systems", Forum on Specification and Design Languages 2008 (FDL'08), September 23-25, 2008, Stuttgart, Germany.
  - (40) Andreas W. Liehr, Klaus J. Buchenrieder, Ulrich Nageldinger: "Visual Feedback for Design-Space Exploration with UML MARTE"; 5th International Conference on Innovations in Information Technology (Innovations'08); Al Ain, December 2008.
  - (41) Kim Grüttner, Frank Oppenheimer, Wolfgang Nebel, Jan Freuer, Joachim Gerlach: „Rapid Prototyping und Synthese eines videobasierten Fahrassistenzsystems mit C++ und SystemC™“, to be published at 10. Symposium Automatisierungs-, Assistenzsysteme und eingebettete Systeme für Transportmittel (AAET'09), 11.-12. Februar 2009, Braunschweig, Germany
  - (42) Jan-Hendrik Oetjens, Ralph Görge, Joachim Gerlach, Wolfgang Nebel: „An Automated Flow for Integrating Hardware IP into the Automotive Systems Engineering Process“, to be published at Design Automation and Test in Europe (DATE) Conference. April 20-24, 2009, Nice, France.
  - (43) Andreas W. Liehr, Klaus J. Buchenrieder, Heike S. Rolfs, Ulrich Nageldinger: „Generation of MARTE Allocation Models from Activity Threads“; in: "Languages of Embed-
-

- 
- ded Systems and their Applications", Springer Lecture Notes in Electrical Engineering (LNEE); to be published 2009.
- (44) Oliver Bringmann, Joachim Gerlach, Ulrich Nageldinger, Jens Stellmacher, „Modeling, Analysis And Refinement Of Heterogeneous Interconnected Systems Using Virtual Platforms“, Virtual Platform Workshop at DAC 2009, San Francisco, CA, July 29th
  - (45) Liehr, A. W. & Buchenrieder, K. J. „Building Performance Simulation Models from UML MARTE System Descriptions“, In Proceedings of the 12th International Conference On Computer Aided Systems Theory, February 2009, Las Palmas, GC, Spain
  - (46) Liehr, A. W. & Mazanek, S. & Buchenrieder, K. J. & Nageldinger, U. „A Model-Driven Engineering Approach to Derive Simulation Models from UML-Based Hardware Descriptions“, In Proceedings of the 20th IASTED International Conference on Modeling and Simulation (MS09), July 2009, Banff, Canada.
  - (47) Timo Schönwald, Jochen Zimmermann, Oliver Bringmann, Wolfgang Rosenstiel: „Network-on-Chip Architecture Exploration Framework“; Euromicro Conference on Digital System Design (DSD), August 2009, Patras, Greece
  - (48) Alexander Viehl, Michael Pressler, Oliver Bringmann; „Bottom-Up Performance Analysis Considering Time Slice Based Software Scheduling at System Level“; International Conference on Hardware/Software Codesign and System Synthesis, October 2009, Grenoble, France
  - (49) Alexander Viehl, Michael Pressler, Oliver Bringmann, Wolfgang Rosenstiel; „White Box Performance Analysis Considering Static Non-Preemptive Software Scheduling“; IEEE/ACM conference on Design, automation and test in Europe (DATE), April 2009, Nice, France
  - (50) Alexander Viehl, Jordan Dukadinov, Oliver Bringmann, Wolfgang Rosenstiel; „TRANSYSCTOR: A General Methodology and Framework for Rule-Based Transformation and Refactoring of SystemC Designs“; 15th Workshop on Synthesis And System Integration of Mixed Information technologies (SASIMI), March 2009, Okinawa, Japan
  - (51) Matthias Krause, Dominik Englert, Oliver Bringmann and Wolfgang Rosenstiel; „Combination of instruction set simulation and abstract RTOS model execution for fast and accurate target software evaluation “; 6th International Conference on Hardware/Software Codesign and System Synthesis (CODES-ISSS), October 2008, Atlanta, USA

---

## 6 Quellenverzeichnis

- [Acce 04] "Property Specification Language Reference Manual V1.1", Accellera Organization, Inc., June, 2004
- [Acce 04] "SystemVerilog 3.1a Language Reference Manual - Accellera's Extensions to Verilog", Accellera Organization, 2004.
- [AISI 04] K. Albers, F. Slomka, "An Event Stream Driven Approximation for the Analysis of Real-Time Systems", 16th Euromicro Conference on Real-Time Systems (ECRTS) 2004, June 30-July 2, Catania, Italy.
- [AnPu 01] J.-L. Anciano, R. Puigjaner: "Annotating UML Diagrams to Carry Out Performance Analysis". In: Proceedings Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'01) (2001) 215-222.
- [Auto 04] <http://www.autosar.org>
- [BaEZ 00] M. Bauer, W. Ecker, A. Zinn: "Graphische Spezifikation und Analyse funktionaler Testabläufe mit MSCs der UML". GI/ITG Workshop „Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen“, Frankfurt, Deutschland, Februar 2000.
- [BaHa 04] N. Bannow, K. Haug: "Performance Analysis and Automated C++ Modularization with the Usage of Module-Adapters for SystemC", Forum on Specification & Design Languages (FDL) 2004, September 14-17, 2004, Lille, France.
- [Bair 03] M. Baird (ed.): "SystemC 2.0.1 Language Reference Manual", Open SystemC Initiative, San Jose, CA, 2003.
- [BeMi 02] L. Benini, G. De Micheli: "Network on Chips: A New SoC Paradigm". IEEE Computer, pp. 70-78, January 2002.
- [Bort 00] U. Bortfeld and C. Mielenz: "Whitepaper C++ System Simulation Interfaces", Internal Report, Infineon Technologies AG, July 2000.
- [BoTh 01] J.-Y. le Boudec, P. Thiran: "Network Calculus - A Theory of Deterministic Queuing Systems for the Internet". Springer, 2001.
- [BGRS 05] A. Braun, J. Gerlach, W. Rosenstiel, A. Siebenborn, O. Bringmann: "SystemC-Based Communication and Performance Analysis", Forum on specification and Design Languages (FDL) Lausanne, Switzerland, 2005.
- [BrSR 05] O. Bringmann, A. Siebenborn, W. Rosenstiel: "Conflict Analysis in Multiprocess Synthesis for Optimized System Integration", IEEE International Conference on Hardware - Software Codesign and System Synthesis, New York, USA, 2005.
- [BSSH 04] A. G. Braun, T. Schubert, M. Stark, K. Haug, J. Gerlach, W. Rosenstiel: "A Case Study: SystemC-Based Design of an Industrial Exposure Control Unit", in C. Grimm (ed.): Languages for System Specification, Chapter 8 (pp. 119-132), Kluwer Academic Publishers, 2004.
- [Burn 95] A. Burns, A. Wellings: "HRT-Hood: A Structured Design Method for hard Real-Time Ada", Systems Elsevier Science B.V, 1995.
- [Can 03] ISO/PRF 11898:2003: "Controller area network (CAN) standard", 2003.
- [ChKT 03] S. Chakraborty, S. Künzli und L. Thiele: "A General Framework for Analysing System Properties in Platform-Based Embedded System Designs". In: Proceedings of Design, Automation and Test in Europe (DATE), Munich, 2003.

- 
- [ChSg 02] R. Chen, M. Sgroi, L. Lavagno, G. Martin, A. Sangiovanni-Vincentelli, J. Rabaey: "Embedded System Design Using UML and Platforms", Forum on Specification & Design Languages (FDL), Marseille, France 2002.
- [Cowa 04] <http://www.coware.com>
- [DöLJ 03] R. Dölling, M. Labbé, P. Jores: "InSiMS (Integrated Co-Simulation for Mixed-Signal-Systems)", Analog 2003, September 2003, Heilbronn, Germany.
- [DöMi 04] R. Dölling, H. Mielenz: "Automatic Identification Method for Analog and Mixed Analog/Digital ICs in Automotive Electronics", International Automotive Conference (IAC) 2004, June 15-16, 2004, Stuttgart, Germany.
- [Flex 04] <http://www.flexray.com>
- [Gerl 04a] J. Gerlach: "Simulink-basierte Modellierung und Synthese von digitalen Systemen der Automobilelektronik", 3. EkompasS Workshop „Entwurfplattformen komplexer angewandter Systeme und Schaltungen“, Mai 18-19, 2004, Hannover, Germany.
- [Gerl 05a] J. Gerlach, „Embedding the Digital Hardware Implementation Flow into a Simulink Based System Design Methodology“, 3rd Open SpeAC Workshop, June 22, 2005, Friedrichshafen, Germany.
- [Gerl 05b] J. Gerlach, „Einbettung des digitalen Designflows in eine Simulink-basierte Entwurfsmethodik“, 4. EkompasS Workshop „Entwurfplattformen komplexer angewandter Systeme und Schaltungen“, April 26-27, 2005, Hannover, Germany.
- [Gerl 04b] J. Gerlach, „Methodik und Werkzeugumgebung zum Simulink-basierten Entwurf von automobilelektronischen Systemen“, Fach- und Kooperationsworkshop „Matlab-Integration und Performanzanalyse“, December 12, 2004, Dresden, Germany.
- [GNOS 03] E. Grimpe, W. Nebel, F. Oppenheimer, T. Schubert: "Object-Oriented Hardware Design and Synthesis Based on SystemC 2.0", In SystemC : Methodologies and Applications, pp. 217-246, Kluwer Academic Publishers, 2003.
- [ITU 99] ITU-T Z.100 (11/99), ITU: "Specification and Description Language (SDL) ", Nov. 1999.
- [Jaco 92] I. Jacobson, et al.: "Object-Oriented Software Engineering: A Use-Case Driven Approach", Reading, MA, Addison-Wesley, 1992.
- [Jaco 99] I. Jacobson, G. Booch, J. Rumbaugh: "The Unified Software Development Process", Addison-Wesley, 1999.
- [Harb 04] K. Harbich: "Betriebsfehleremulation für sicherheitskritische Automotive-Applikationen", In Design & Verification – Automotive, publish-industry Verlag, September 2004.
- [KeNe 04] M. Kersten, W. Nebel: "On Detecting Deadlocks in Large UML Models - based on an Expressive Subset", IFIP Working Conference on Distributed and Parallel Embedded Systems (DIPES) 2004, August 23-26, 2004, Toulouse, France.
- [Kuhn 01] T. Kuhn, T. Oppold, M. Winterholer, W. Rosenstiel, M. Edwards, Y. Kashai: "A Framework for Object Oriented Hardware, Specification, Verification, and Synthesis". Proceedings of Design Automation Conference (DAC), 2001.
- [Kuma 03] S. Kumar: "On Packet Switched Networks for On-Chip Communication". In: A. Jantsch, H. Tenhunen: Networks-on-Chip, Kluwer Academic Publishers, 2003.
- [LaSc 04] L. Lavagno, F. Schäfer: "UML Platform", Workshop on UML to SystemC System Design Flow, Lille, France, May 2004.
-

- 
- [LaMa 03] L. Lavagno, G. Martin, B. Selic: "UML for Real: Design of Embedded Real-Time Systems, Kluwer Academic Pub, September 2003.
- [LaPa 05] L. Lavagno, C. Passarone, V. Shah, Y. Watanabe: "A Time Slice based Scheduler Model for System Level Design", DATE Munich, Germany, 2005
- [Lazo 84] E. D. Lazowska, J. L. Zahorjan, G. S. Graham, K. C. Sevcik, K.C: "Quantitative System Performance: Computer System Analysis Using Queuing Network Models", Prentice Hall, 1984.
- [LeLeSo 05] R. Leveugle, A. Lemarchal, L. Sourgen : Special Session on Secure Implementations, Proceedings of the 11th International On-Line Testing Symposium, Saint Raphael 2005, IEEE Computer Society, pp.115-117.
- [MaLa 01] G. Martin, L. Lavagno, J. Louis-Guerin, Embedded UML: "A merger of real-time UML and co-design, Proceedings of CODES 2001, Copenhagen, April 2001.
- [MLPB 00] M.d. Miguel, T. Lambolais, S. Piekarec, S. Betgé-Brezetz, J. Péquery: "Automatic Generation of Simulation Models for the Evaluation of Performance and Reliability of Architectures Specified in UML". In: Engineering Distributed Objects, Second International Workshop, EDO 2000. Davis, CA, USA (2000).
- [MiKe 03] A. Mihal, K. Keutzer: "Mapping Concurrent Application onto Architectural Platforms". In: A. Jantsch, H. Tenhunen: Networks-on-Chip, Kluwer Academic Publishers, 2003.
- [Most 04] <http://www.mostnet.org/>
- [MüSI 03] DE10057651C2. R. Münzenberger, R. Slomka, M. Dörfel, O. Bringmann: "Verfahren zur Herstellung von computergestützten Echtzeitsystemen". Deutsches Patent DE10057651C2, Priorität aus DE10044021 vom 6.9.2000, veröffentlicht am 22.5.2003, erteilt am 22.5.2003, Patentklasse (IPC7) G06F17/50.
- [NWSE 05] U. Nageldinger, Th. Wilde, R. Schwencker, J. Eckmüller: "A SystemC-based Modeling Methodology for complex ICs"; MEDEA+ DAC 2005, Mesnuls, Frankreich, Juni 2005.
- [OeGR 04] J.H. Oetjens, J. Gerlach, W. Rosenstiel: "An XML Based Approach for Flexible Representation and Transformation of System Descriptions", Forum on Specification & Design Languages (FDL) 2004, September 14-17, 2004, Lille, France.
- [OMG 2003] "UML Profile for Schedulability, Performance, and Time Specification". Version 1.0 OMG, September 2003. [http://www.omg.org/technology/documents/modeling\\_spec\\_catalog.htm](http://www.omg.org/technology/documents/modeling_spec_catalog.htm)
- [Oppe 04] F. Oppenheimer: "OOCOSIM - An Object-Oriented Codesign Method for Embedded HW/SW Systems", International Conference on Design, Automation and Test in Europe (DATE) 2004, PhD Forum, February 16-20, 2004, Paris, France.
- [OpZN 01] Frank Oppenheimer, Dongming Zhang, Wolfgang Nebel: "Modelling Communication Interfaces with ComiX", Proceedings of the 6th International Conference on Reliable Software Technologies - Ada-Europe 2001, Springer, 2001.
- [PCT 02] Patent Cooperation Treaty (PCT). Erfindung: "Verfahren zur Herstellung von computergestützten Echtzeitsystemen". Weltorganisation für geistiges Eigentum, Internationale Veröffentlichungsnummer WO 02/21261 A2, 14.03.2002.
- [PoEP 00] P. Pop, P. Eles und Z . Peng: "Performance Estimation for Embedded Systems with Data and Control Dependencies", In: Proceedings of CODES, 2000.
- [RZJE 02] K. Richter, D. Ziegenbein, M. Jersak und R. Ernst: "Model Composition for Scheduling Analysis in Platform Design", In: Proceedings of DAC, 2002.
-

- 
- [ScBR 05] J. Schnerr, O. Bringmann, W. Rosenstiel: "Cycle Accurate Binary Translation for Simulation Acceleration in Rapid Prototyping of SoCs", Design, Automation and Test in Europe (DATE), Munich, 2005.
- [SchR 03] J. Schnerr, G. Haug, W. Rosenstiel: "Instruction set emulation for rapid prototyping of SoCs, Design", Automation and Test in Europe (DATE), Munich, 2003.
- [ScRo 04] S. Schmitt, W. Rosenstiel: "Verification of a Microcontroller IP Core for SoC Designs Using Low-Cost Prototyping Environments", Designers Forum, DATE, 2004.
- [SHGA 04] T. Schubert, J. Hanisch, J. Gerlach, J.-E. Appell, W. Nebel: "Evaluation of a Refinement-Driven SystemC-Based Design Flow", Design Automation and Test in Europe (DATE) 2004, Designer's Forum, February 17-19, 2004, Paris, France.
- [SiSg 00] J. da Silva Jr., M. Sgroi, F. De Bernardinis, S.F Li, A. Sangiovanni-Vincentelli and J. Rabaey: "Wireless Protocols Design: Challenges and Opportunities", Proceedings of the 8th IEEE International Workshop on Hardware/Software Codesign, CODES '00, San Diego, CA, USA, May 2000.
- [SiBR 04a] A. Siebenborn, O. Bringmann, und W. Rosenstiel: "Communication Analysis for System on Chip Design", Design, Automation and Test in Europe (DATE), Paris, 2004.
- [SiBR 04b] A. Siebenborn, O. Bringmann, W. Rosenstiel: "Communication Analysis for Network-on-Chip", International Conference on Parallel Computing in Electrical Engineering (PARELEC), Dresden, 2004.
- [SiBR 02] A. Siebenborn, O. Bringmann und W. Rosenstiel: "Worst-case performance analysis of parallel, communicating software processes", International Symposium on Hardware/Software Co-Design (CODES), Estes Park, USA, 2002.
- [SiTe 03] Frank Slomka, Jürgen Teich: "A Model for Buffer Exploration in EDF Scheduled Embedded Systems", In: 11. EIS-Workshop, Entwurf Integrierter Schaltungen und Systeme, Erlangen. Hrsg. v. VDE/VDI Gesellschaft Mikroelektronik, Mikro- und Feinwerktechnik (GMM), Berlin: VDE-Verlag, GMM-Fachbericht, 2003, pp. 91-96.
- [SWSK 04a] C. Schulz-Key, M. Winterholer, T. Schweizer, T. Kuhn, W. Rosenstiel: "Object-Oriented Hardware Design and Synthesis with SystemC". Forum on Specification & Design Languages (FDL), Lille, France, 2004.
- [SWSK 04b] C. Schulz-Key, M. Winterholer, T. Schweizer, T. Kuhn, and W. Rosenstiel: "Object-Oriented Modeling and Synthesis of SystemC Specifications". Proceedings of the Asia South Pacific Design Automation Conference (ASPDAC'04), Yokohama, Japan, 2004.
- [SoHe 03] J.-P. Soininen, H. Heusala: "A Design Methodology for NoC-Based Systems". In A. Jantsch, H. Tenhunen: Networks-on-Chip, Kluwer Academic Publishers, 2003.
- [Vast 04] <http://www.vastsystems.com/>
- [Veri 02] "e Language Reference Manual", Verisity Ltd., 2002.
- [Vect 04a] <http://www.vector-informatik.de/deutsch/produkte/canalyzer.html>
- [Vect 04b] <http://www.vector-informatik.de/deutsch/produkte/davinci.html>
- [ViBR 05] A. Viehl, O. Bringmann, W. Rosenstiel: Performance Analysis of Sequence Diagrams for SoC Design, 2nd UML for SoC Design Workshop at 42nd Design Automation Conference (DAC), Anaheim, California, 2005.
- [VISION] Homepage des Projektes VISION, <http://www.edacentrum.de/vision>
-

- [WSKR 02] M. Winterholer, C. Schulz-Key, T. Kuhn, W. Rosenstiel: "Object-Oriented Synthesis, Modelling and Partitioning for SoC Design". Proceedings of International Workshop on IP-Based SoC Design, Grenoble, France, October, 2002.
- [XiWo 01] Y. Xie, W. Wolf: "Allocation and Scheduling of Conditional Task Graph in Co-Synthesis", In: Proceedings of Design, Automation and Test in Europe (DATE), Munich, 2001.
- [XuLe 04] Z. Xu, A. Lehmann: "Generating Queuing Network Models from UML-based Models for Software Performance Prediction", Accepted by the 2004 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2004). San Jose, California, USA (2004)
- [XuLL 03] Z. Xu, J. Lüthi, A. Lehmann: "Predicting Software Performance Based on UML Models during the Unified Software Development Process", In: Proceedings of the Critical Systems Development with UML (CSDUML) Workshop in the Sixth International Conference on the Unified Modeling Language (UML 2003). San Francisco, USA (2003) (TUM-INFO-09-I0323-80/1.-FI)
- [XuLe 02] Z. Xu, A. Lehmann: "Automated Generation of Queuing Network Model from UML-based Software Models with Performance Annotations", Technischer Bericht Nr. 2002-06, Universität der Bundeswehr München, Fakultät für Informatik.
- [Xu 04] Z. Xu: "Software Performance Prediction as Part of the Unified Software Development Process", Dissertation, Universität der Bundeswehr München, Institut für Technische Informatik, 2004.
- [YeWo 98] T.-Y. YEN, W. Wolf: "Performance Estimation for Real-Time Distributed Embedded Systems", In: IEEE Transactions on Parallel and Distributed Systems, Bd. 9, 1998.

## 7 Tabellenverzeichnis

Tabelle 1: Erwartete Ergebnisse von VISION hinsichtlich der Entwurfsmöglichkeit	8
Tabelle 2: Erzielte Ergebnisse von VISION hinsichtlich der Entwurfsmöglichkeit	70

---

## 8 Abbildungsverzeichnis

Abbildung 1: Vernetzte automobilelektronische Systemfunktionen	6
Abbildung 2: Verspätete Markteinführung und Umsatzeinbußen aufgrund später Fehleraufdeckung (Quelle: Integrated Communications Design May, 2001)	7
Abbildung 3: Projektstruktur und Arbeitspakete	10
Abbildung 4: Kooperationen mit Projekten und Organisationen	18
Abbildung 5: Kooperationen mit Firmen	19
Abbildung 6: Vorgeschlagene Anpassung von in XML und/oder UML beschriebenen Modelldarstellungen, um eine flexible und optimierte Repräsentation im Metamodell gewährleisten und gleichzeitig einen Codegenerator bedienen zu können.	22
Abbildung 7: Modellierung und Abbildung von Plattformkomponenten	24
Abbildung 8: Verfeinerung der Systemmodells und Generierung eines Virtuellen Prototyps	25
Abbildung 9: Eclipse-basierte Plattformmodellierung	26
Abbildung 10: Generierungsflow basierend auf Beschreibungsmethodik	28
Abbildung 11: Modellierungswerkzeug für Netzwerkarchitekturen	29
Abbildung 12: Tool-Flow	30
Abbildung 13: Systementwurfsmethodik für FlexRay-Szenarien	31
Abbildung 14: Zusammenstellung und Aufbau einer Kommunikationstopologie aus generischen Komponenten	32
Abbildung 15: Simulationsumgebung für eine host code Ausführung des Systemverhaltens, hier an Hand des Beispiels für einen Bremsknoten	33
Abbildung 16: Vereinfachte Struktur eines abstrakten FlexRay Knotens	34
Abbildung 17: Simulationsumgebung für ein aus zwei Knoten bestehendes Kommunikationsmodell	34
Abbildung 18: Modell des abstrakten Kommunikationsverfahrens	36
Abbildung 19: Verfeinerungsstufen der Kommunikationsverfeinerung	37
Abbildung 20: Architektur einer Schnittstellenanpassung	39
Abbildung 21: Kontrollfluss-orientierte Spezifikation generischer Kommunikationsprotokolle	39
Abbildung 22: Beispiel Kommunikations-Abhängigkeits-Graph aus Arbeitspaket 2	40
Abbildung 23: Konfliktfreie Abbildung von Kommunikationen auf die Architektur	41
Abbildung 24 – Abbildung des BBW Anwendungsbeispiels auf verschiedene Kommunikations-Topologien	43
Abbildung 25: Durch eine partielle Realisierung der Funktionen in HW Modulen (in SystemC mit annotiertem Zeitverhalten) und der Ausführung der verbleibenden Funktionen auf dem Prozessormodell können unterschiedliche Varianten des HW/SW mapping simulativ bewertet werden.	44
Abbildung 26: Erweiterter Analyseablauf	45
Abbildung 27: Parametrisierung der Abbildung kommunizierender Prozesse	47
Abbildung 28: Framework zur Generierung und Auswertung von Systemmodellen zur Performanzanalyse	48

---

Abbildung 29: Erzeugung von Komponenten-Modellen	49
Abbildung 30: Gateway-Architektur zur Kopplung von Subnetzdomänen	50
Abbildung 31: Simulations- und Analyseumgebung	51
Abbildung 32: M_CAN Controller Architektur (a) und Systemeinbindung (b)	52
Abbildung 33: SystemC-basierte Verifikationsmethodik	53
Abbildung 34: Anbindung von Applikationssoftware	54
Abbildung 35: Verifikationsumgebung	55
Abbildung 36: Schichten des "Layered Modelling Approach"	57
Abbildung 37: Anwendung des "Layered Modelling Approach"	58
Abbildung 38: XML/XSLT-basierter Transformationsansatz	59
Abbildung 39: Low-Power Transformation basierend auf Clock-Gating	59
Abbildung 40: Benutzer-geführte Spezifikation von Transformationsregeln	59
Abbildung 41: Transformation zur Erweiterung von Busstrukturen um Fehlererkennungs- und -korrekturmechanismen	60
Abbildung 42: Schichten-/Säulenmodell für den Entwurf verlässlicher Kommunikationsarchitekturen	62
Abbildung 43: Galois Counter Mode Einheit des AES-128 Verschlüsselungsverfahrens	62
Abbildung 44: Applikationsspezifische Anpassung in VISION	63