

Vorlesung

Automotive Software Engineering

Teil 8-2 Beispiele aus der Praxis

Sommersemester 2014

Prof. Dr. rer. nat. Bernhard Hohlfeld

Bernhard.Hohlfeld@mailbox.tu-dresden.de

Technische Universität Dresden, Fakultät Informatik
Honorarprofessur Automotive Software Engineering

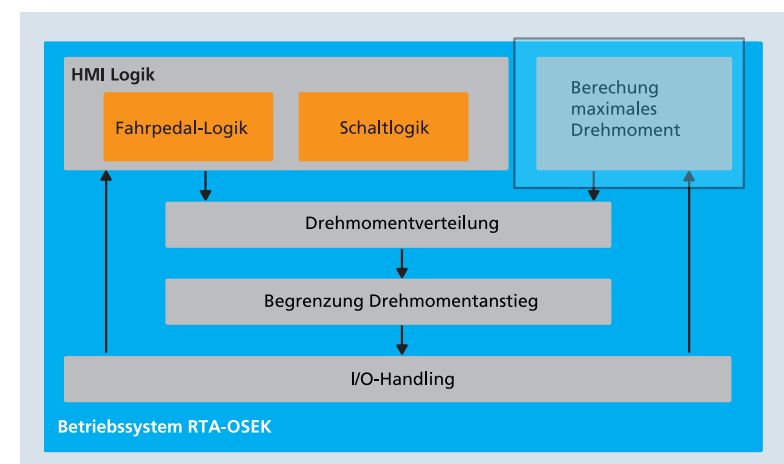
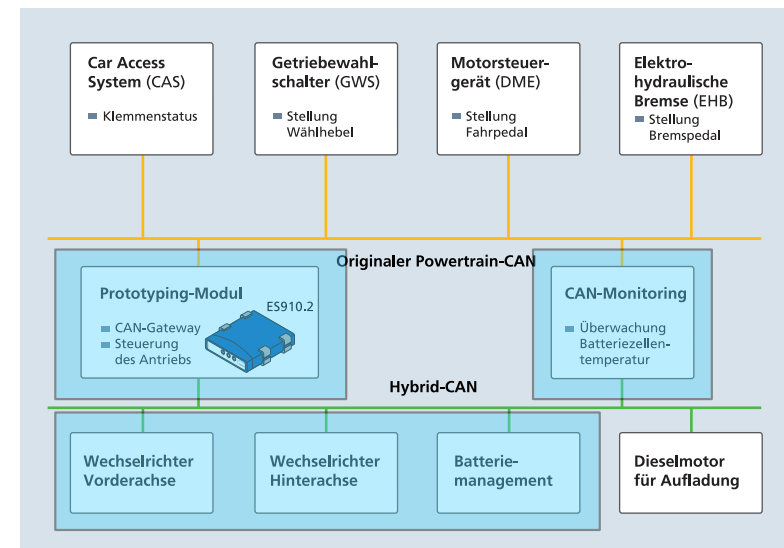
Entwicklung einer Antriebssteuerung für ein Hybridfahrzeug in einer Rapid Prototyping-Umgebung

1. Einleitung
2. Das Fahrzeug
3. Die Bordnetz-Architektur
- 4. Funktionsentwicklung mit ASCET**
 1. Fahrpedal-Logik
 - 2. Berechnung des maximalen Drehmoments**
 3. Drehmomentverteilung
 4. Drehmomentanstiegsbegrenzung
 5. Funktionsmodellierung
5. Die Erprobung
 1. Offline-Simulation
 2. Rapid Prototyping
6. Höhere Antriebsfunktion
7. Fazit
8. Abkürzungsverzeichnis

Quelle: http://www.etas.com/de/products/download_center.php?entrylist=13992

Berechnung des maximalen Drehmoments

- Die wassergekühlten Wechselrichter und Motoren sowie die Batterien werden temperaturüberwacht. Um ein Überhitzen der Komponenten zu vermeiden, werden in Abhängigkeit von den Temperaturen, die maximalen Drehmomentwerte für beide Achsen ermittelt. Hierzu werden, für jede Achse getrennt, entsprechende Kennlinien verwendet. Ein weiterer Einflussfaktor ist der Ladezustand der Batterie. Durch die Begrenzung des Drehmomentwertes wird ein Überladen bzw. Tiefentladen der Batterie verhindert.



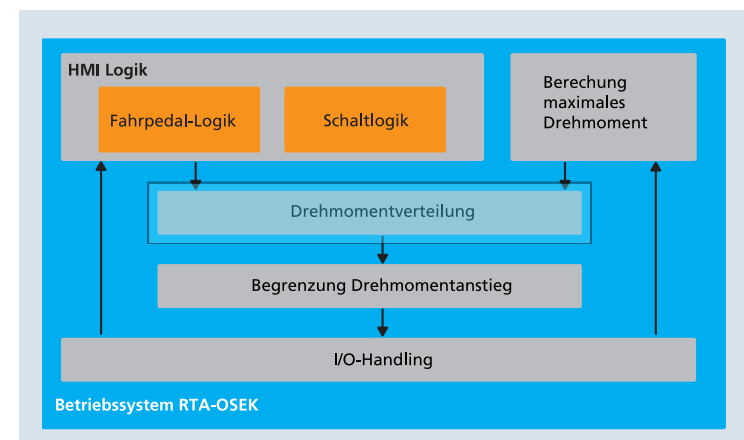
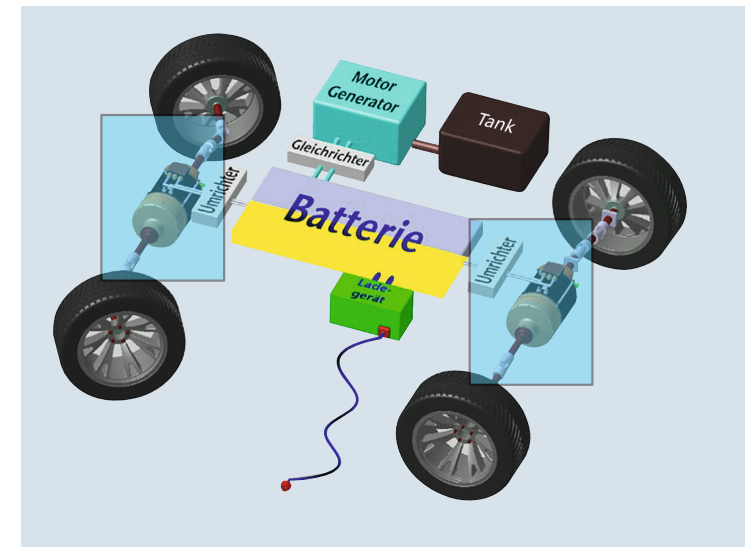
Entwicklung einer Antriebssteuerung für ein Hybridfahrzeug in einer Rapid Prototyping-Umgebung

1. Einleitung
2. Das Fahrzeug
3. Die Bordnetz-Architektur
- 4. Funktionsentwicklung mit ASCET**
 1. Fahrpedal-Logik
 2. Berechnung des maximalen Drehmoments
 - 3. Drehmomentverteilung**
 4. Drehmomentanstiegsbegrenzung
 5. Funktionsmodellierung
5. Die Erprobung
 1. Offline-Simulation
 2. Rapid Prototyping
6. Höhere Antriebsfunktion
7. Fazit
8. Abkürzungsverzeichnis

Quelle: http://www.etas.com/de/products/download_center.php?entrylist=13992

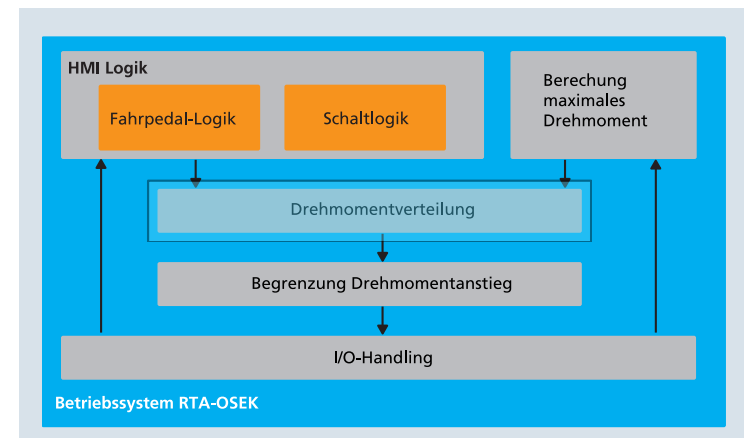
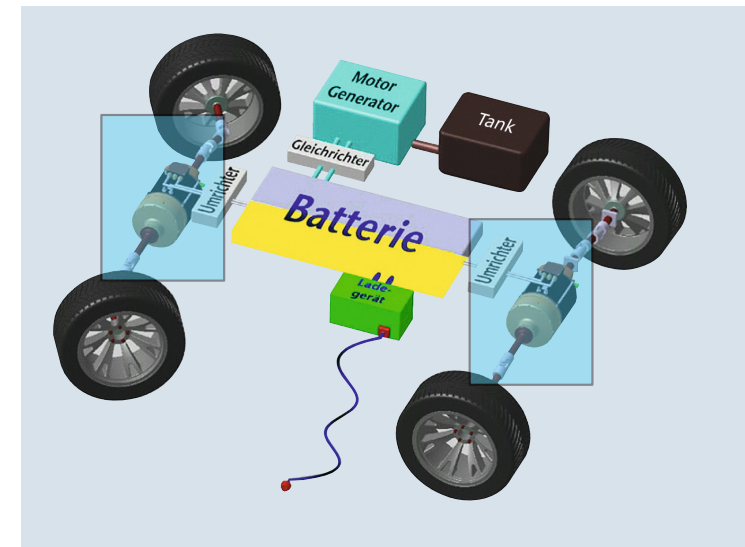
Drehmomentverteilung (1)

- Nachdem das Soll-Drehmoment ermittelt worden ist, muss dieses auf die beiden Achsen verteilt werden. Die Verteilung erfolgt dabei variabel nach folgender Strategie:
- Zuerst wird das Drehmoment in Abhängigkeit von der Geschwindigkeit auf die beiden Achsen verteilt. Dies geschieht unter Verwendung einer Kennlinie mit mehreren Stützstellen und linearer Interpolation.
- Die Soll-Drehmomentvorgaben werden anschließend für die Vorder- und die Hinterachse mit den ermittelten Grenzwerten verglichen. Strategie ist es, das vom Fahrer angeforderte Gesamtdrehmoment soweit wie möglich aufzubauen. Sollte eine Drehmomentvorgabe an einer Achse den



Drehmomentverteilung (2)

- Die Soll-Drehmomentvorgaben werden anschließend für die Vorder- und die Hinterachse mit den ermittelten Grenzwerten verglichen. Strategie ist es, das vom Fahrer angeforderte Gesamtdrehmoment soweit wie möglich aufzubauen. Sollte eine Drehmomentvorgabe an einer Achse den Grenzwert überschreiten, so wird das Drehmoment an dieser Achse begrenzt. Es wird jedoch versucht dies auszugleichen, indem das Drehmoment an der anderen Achse soweit wie möglich (im Rahmen der Grenze) heraufgesetzt wird, um dem Fahrerwunsch nachzukommen. Die ideale Achsverteilung wird also zu Gunsten des Drehmomentwunsches aufgegeben.



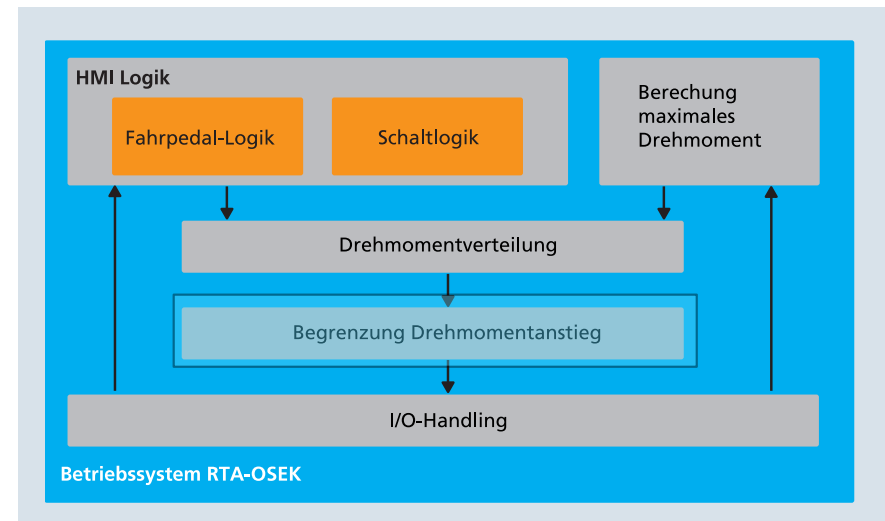
Entwicklung einer Antriebssteuerung für ein Hybridfahrzeug in einer Rapid Prototyping-Umgebung

1. Einleitung
2. Das Fahrzeug
3. Die Bordnetz-Architektur
- 4. Funktionsentwicklung mit ASCET**
 1. Fahrpedal-Logik
 2. Berechnung des maximalen Drehmoments
 3. Drehmomentverteilung
 - 4. Drehmomentanstiegsbegrenzung**
 5. Funktionsmodellierung
5. Die Erprobung
 1. Offline-Simulation
 2. Rapid Prototyping
6. Höhere Antriebsfunktion
7. Fazit
8. Abkürzungsverzeichnis

Quelle: http://www.etas.com/de/products/download_center.php?entrylist=13992

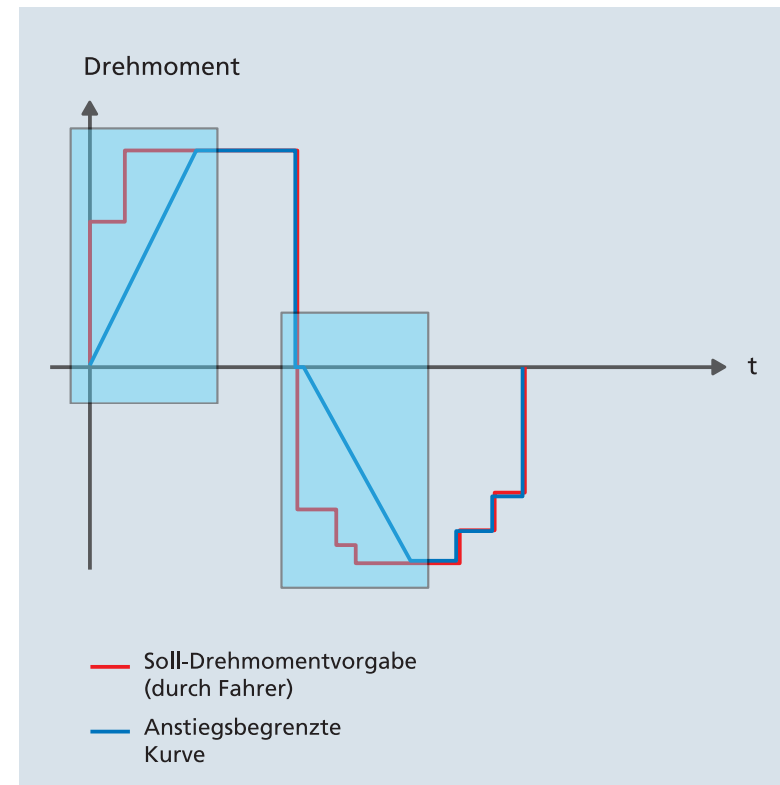
Drehmomentanstiegsbegrenzung (1)

- Elektromotoren können innerhalb sehr kurzer Zeit ein sehr hohes Drehmoment aufbringen. Um eine Beschädigung des Antriebsstrangs zu vermeiden, wird die auf die Antriebswellen wirkende Drehmomentänderung (1. Ableitung des Drehmoments nach der Zeit) betragsmäßig auf MAX_TRQ_CH [Nm/s] begrenzt. Dazu wird das durch die Elektromotoren zu erzeugende Drehmoment mit einer Rampenfunktion aufgebaut. Ein sofortiges Abfallen des Drehmoments auf einen betragsmäßig niedrigeren Wert mit gleichem Vorzeichen bzw. auf 0 Nm kann zu keiner Beschädigung führen und ist deshalb möglich.



Drehmomentanstiegsbegrenzung (2)

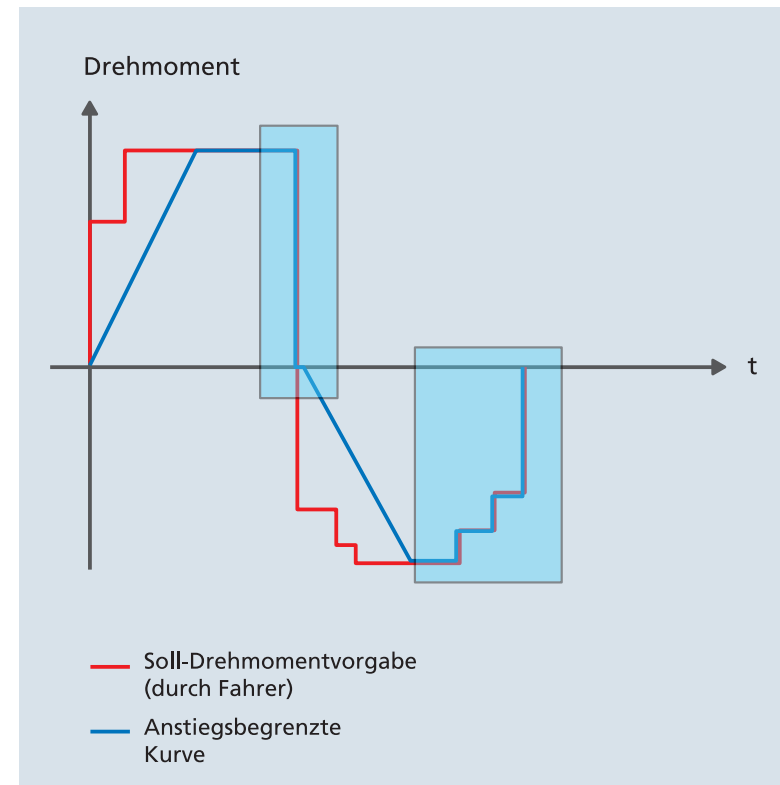
- Elektromotoren können innerhalb sehr kurzer Zeit ein sehr hohes Drehmoment aufbringen. Um eine Beschädigung des Antriebsstrangs zu vermeiden, wird die auf die Antriebswellen wirkende Drehmomentänderung (1. Ableitung des Drehmoments nach der Zeit) betragsmäßig auf MAX_TRQ_CH [Nm/s] begrenzt. Dazu wird das durch die Elektromotoren zu erzeugende Drehmoment mit einer Rampenfunktion aufgebaut. Ein sofortiges Abfallen des Drehmoments auf einen betragsmäßig niedrigeren Wert mit gleichem Vorzeichen bzw. auf 0 Nm kann zu keiner Beschädigung führen und ist deshalb möglich.



Drehmomentanstiegsbegrenzung

Drehmomentanstiegsbegrenzung (3)

- Elektromotoren können innerhalb sehr kurzer Zeit ein sehr hohes Drehmoment aufbringen. Um eine Beschädigung des Antriebsstrangs zu vermeiden, wird die auf die Antriebswellen wirkende Drehmomentänderung (1. Ableitung des Drehmoments nach der Zeit) betragsmäßig auf MAX_TRQ_CH [Nm/s] begrenzt. Dazu wird das durch die Elektromotoren zu erzeugende Drehmoment mit einer Rampenfunktion aufgebaut. Ein sofortiges Abfallen des Drehmoments auf einen betragsmäßig niedrigeren Wert mit gleichem Vorzeichen bzw. auf 0 Nm kann zu keiner Beschädigung führen und ist deshalb möglich.



Drehmomentanstiegsbegrenzung

Entwicklung einer Antriebssteuerung für ein Hybridfahrzeug in einer Rapid Prototyping-Umgebung

1. Einleitung
2. Das Fahrzeug
3. Die Bordnetz-Architektur
- 4. Funktionsentwicklung mit ASCET**
 1. Fahrpedal-Logik
 2. Berechnung des maximalen Drehmoments
 3. Drehmomentverteilung
 4. Drehmomentanstiegsbegrenzung
- 5. Funktionsmodellierung**
5. Die Erprobung
 1. Offline-Simulation
 2. Rapid Prototyping
6. Höhere Antriebsfunktion
7. Fazit
8. Abkürzungsverzeichnis

Quelle: http://www.etas.com/de/products/download_center.php?entrylist=13992

Modellbasierte Entwicklung



Technische
Universität
Braunschweig



Fahrzeuginformatik Modellbasierte Entwicklung

Wintersemester 2012/13

Ina Schaefer

(mit Material von Prof. Dr. B. Rumpe und Dr. M. Trapp)

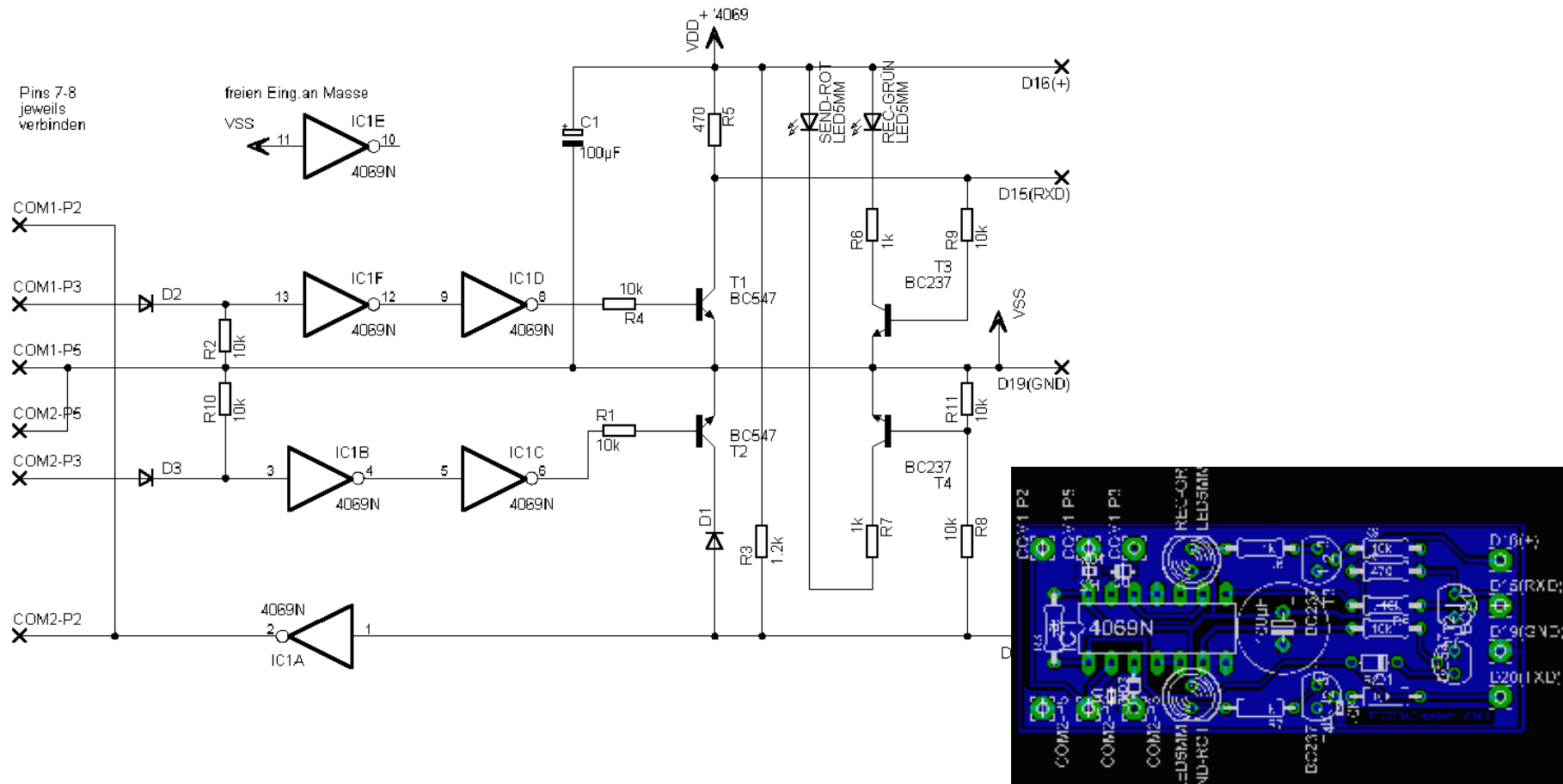
Vorlesung Fahrzeuginformatik

- Entwicklungsprozesse
- Anforderungsmanagement
- Entwicklungsmethodik für verteilte Funktionen
- Echtzeitsysteme
- Vernetzung und Bussysteme
- **Modellbasierte Entwicklung (90 Folien, 2 DS)**
- Qualitätssicherung
- Normen und Standards
- Konfigurations- und Variantenmanagement

Modellbasierte Entwicklung

- Modellbasierte Entwicklung (MBE)
- Modelbased Development (MBD)
- Warum Modellbasierte Entwicklung?
 - Es gibt in unterschiedlichen Domänen wie z. B. den Ingenieursdisziplinen etablierte Beschreibungs- bzw. Modellierungssprachen (oft grafisch)
 - Die Ingenieure können sich in diesen Modellierungssprachen besser verständigen als in Programmiersprachen
 - Über Modelle kann man sicher besser unterhalten wie über Code
 - Bei der werkzeuggestützten Umsetzung der Modelle in Code gibt es potentiell weniger Fehler als bei der manuellen Umsetzung
 - Codegeneratoren werden nicht müde und lassen sich nicht ablenken
 - Eindeutige und einheitliche Interpretation des Modells
 - Bei der werkzeuggestützten Umsetzung der Modelle in Code gibt es potentiell weniger Aufwand als bei der manuellen Umsetzung
 - Modellelemente werden einmal in Code umgesetzt und nicht jedesmal aufs Neue

Elektrotechnik: Schaltkreise nach DIN/ISO-Normen



Der Modellbegriff

- Ein Modell ist seinem Wesen nach eine in Maßstab, Detailliertheit und/oder Funktionalität verkürzte, bzw. abstrahierte Darstellung des originalen Systems. (Stachowiak 1973)
- Ein Modell ist eine vereinfachte, auf ein bestimmtes Ziel hin ausgerichtete Darstellung der Funktion eines Gegenstands oder des Ablaufs eines Sachverhalts, die eine Untersuchung oder eine Erforschung erleichtert oder erst möglich macht. (Balzert 2000)
- Modelle reduzieren Komplexität durch Abstraktion.

Modellbildung - Ein einfaches Beispiel

Modellbildung - Ein einfaches Beispiel

- Fahrzeug A hat einen Tankinhalt von 79 l und eine Reichweite von 920 km.

Modellbildung - Ein einfaches Beispiel

- Fahrzeug A hat einen Tankinhalt von 79 l und eine Reichweite von 920 km.
- Fahrzeug B hat einen Tankinhalt von 42 l und eine Reichweite von 704 km.

Modellbildung - Ein einfaches Beispiel

- Fahrzeug A hat einen Tankinhalt von 79 l und eine Reichweite von 920 km.
- Fahrzeug B hat einen Tankinhalt von 42 l und eine Reichweite von 704 km.

Modellbildung - Ein einfaches Beispiel

- Fahrzeug A hat einen Tankinhalt von 79 l und eine Reichweite von 920 km.
- Fahrzeug B hat einen Tankinhalt von 42 l und eine Reichweite von 704 km.
- Modellbildung?

Modellbildung - Ein einfaches Beispiel

- Fahrzeug A hat einen Tankinhalt von 79 l und eine Reichweite von 920 km.
- Fahrzeug B hat einen Tankinhalt von 42 l und eine Reichweite von 704 km.
- Modellbildung?

Modellbildung - Ein einfaches Beispiel

- Fahrzeug A hat einen Tankinhalt von 79 l und eine Reichweite von 920 km.
- Fahrzeug B hat einen Tankinhalt von 42 l und eine Reichweite von 704 km.
- Modellbildung?
- Kraftstoffverbrauch auf 100 km

Modellbildung - Ein einfaches Beispiel

- Fahrzeug A hat einen Tankinhalt von 79 l und eine Reichweite von 920 km.
- Fahrzeug B hat einen Tankinhalt von 42 l und eine Reichweite von 704 km.
- Modellbildung?
- Kraftstoffverbrauch auf 100 km
 - Abtraktion von

Modellbildung - Ein einfaches Beispiel

- Fahrzeug A hat einen Tankinhalt von 79 l und eine Reichweite von 920 km.
- Fahrzeug B hat einen Tankinhalt von 42 l und eine Reichweite von 704 km.
- Modellbildung?
- Kraftstoffverbrauch auf 100 km
 - Abstraktion von
 - Tankinhalt

Modellbildung - Ein einfaches Beispiel

- Fahrzeug A hat einen Tankinhalt von 79 l und eine Reichweite von 920 km.
- Fahrzeug B hat einen Tankinhalt von 42 l und eine Reichweite von 704 km.
- Modellbildung?
- Kraftstoffverbrauch auf 100 km
 - Abstraktion von
 - Tankinhalt
 - Reichweite

Modellbildung - Ein einfaches Beispiel

- Fahrzeug A hat einen Tankinhalt von 79 l und eine Reichweite von 920 km.
- Fahrzeug B hat einen Tankinhalt von 42 l und eine Reichweite von 704 km.
- Modellbildung?
- Kraftstoffverbrauch auf 100 km
 - Abtraktion von
 - Tankinhalt
 - Reichweite

	Tank (l)	Reich- weite (km)	Verbrauch (l / 100 km)
Fahrzeug A	79	920	8,6
Fahrzeug B	42	704	6,0

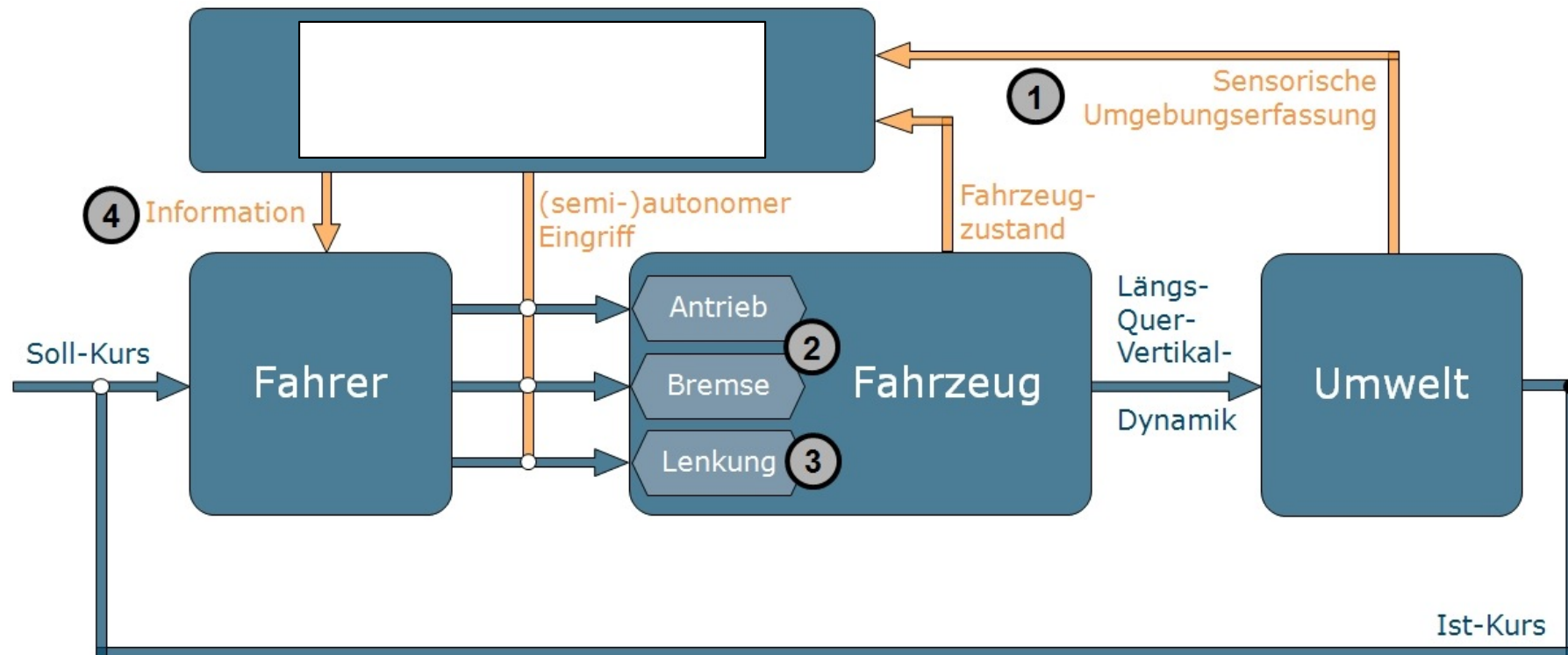
Modellbildung - Ein einfaches Beispiel

- Fahrzeug A hat einen Tankinhalt von 79 l und eine Reichweite von 920 km.
- Fahrzeug B hat einen Tankinhalt von 42 l und eine Reichweite von 704 km.
- Modellbildung?
- Kraftstoffverbrauch auf 100 km
 - Abtraktion von
 - Tankinhalt
 - Reichweite

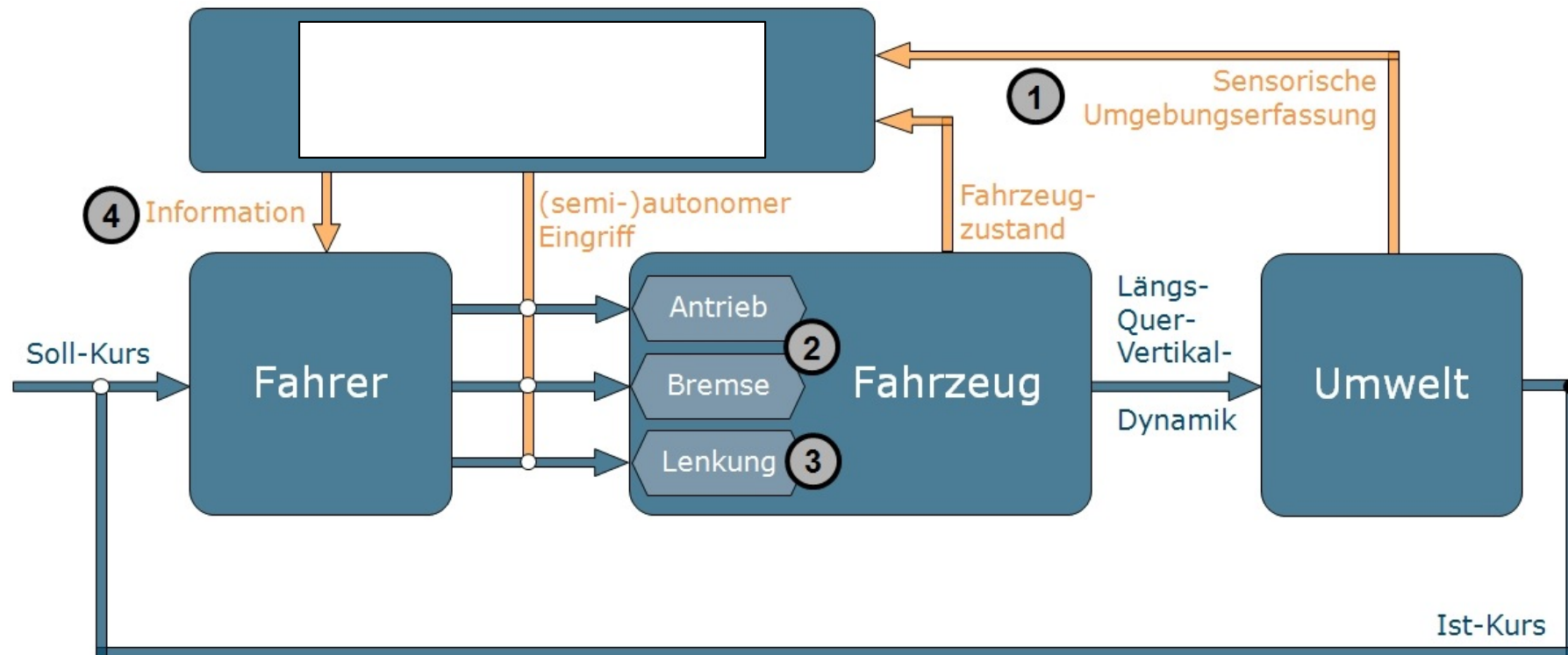
	Tank (l)	Reich- weite (km)	Verbrauch (l / 100 km)
Fahrzeug A	79	920	8,6
Fahrzeug B	42	704	6,0



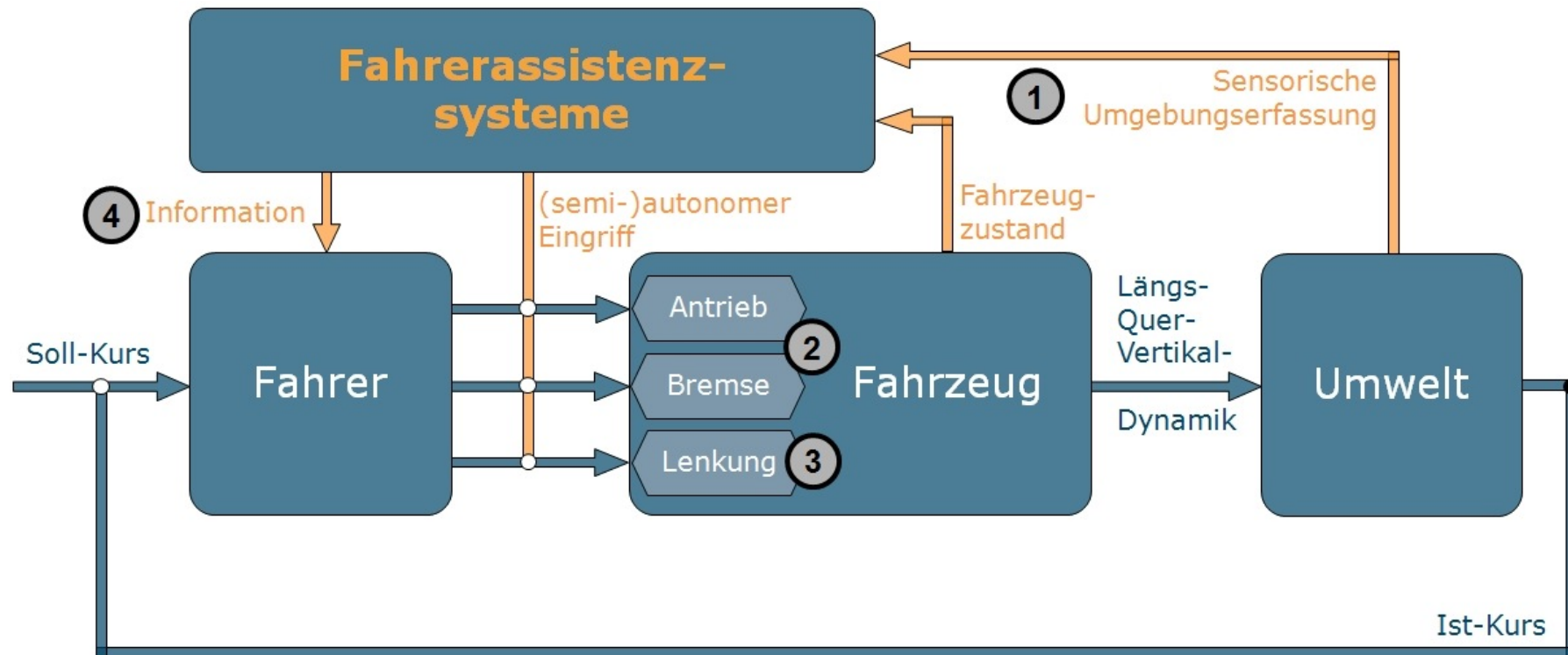
Modell eines ?



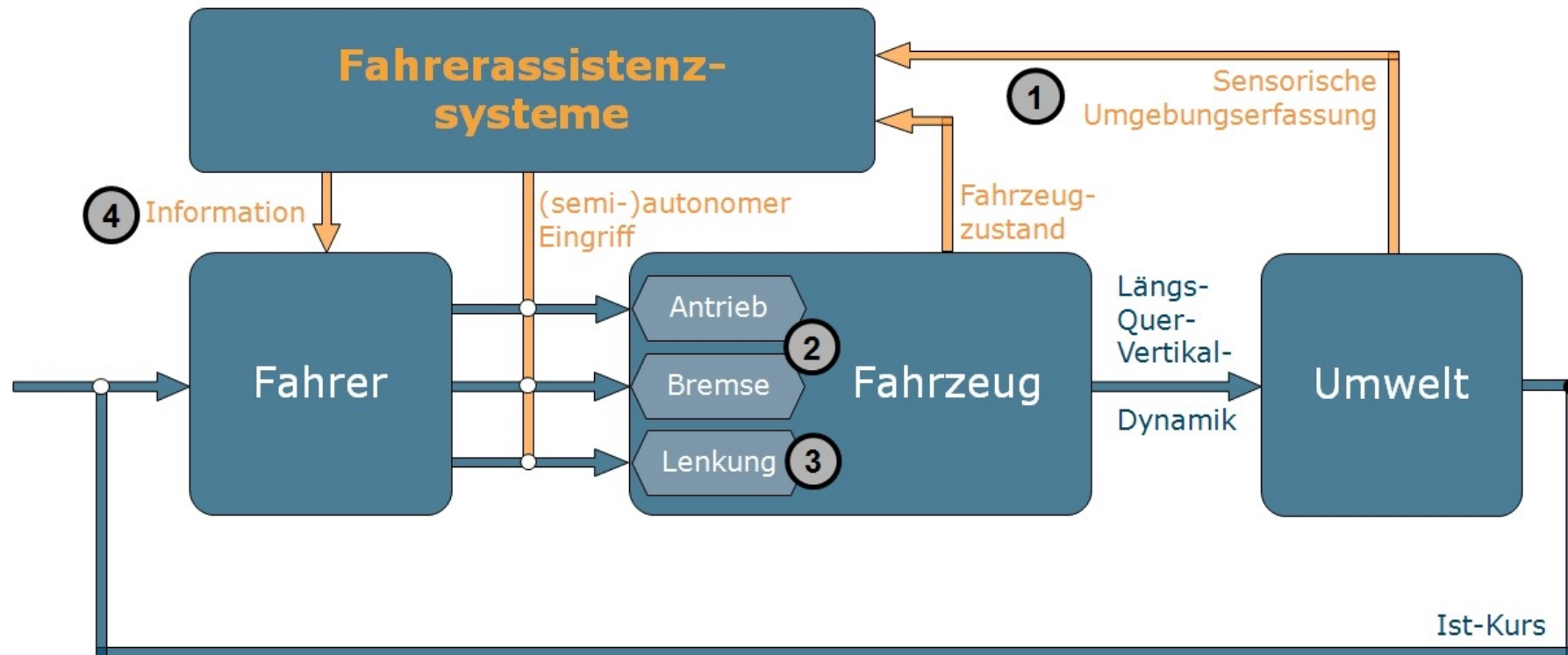
Modell eines Fahrerassistenzsystems



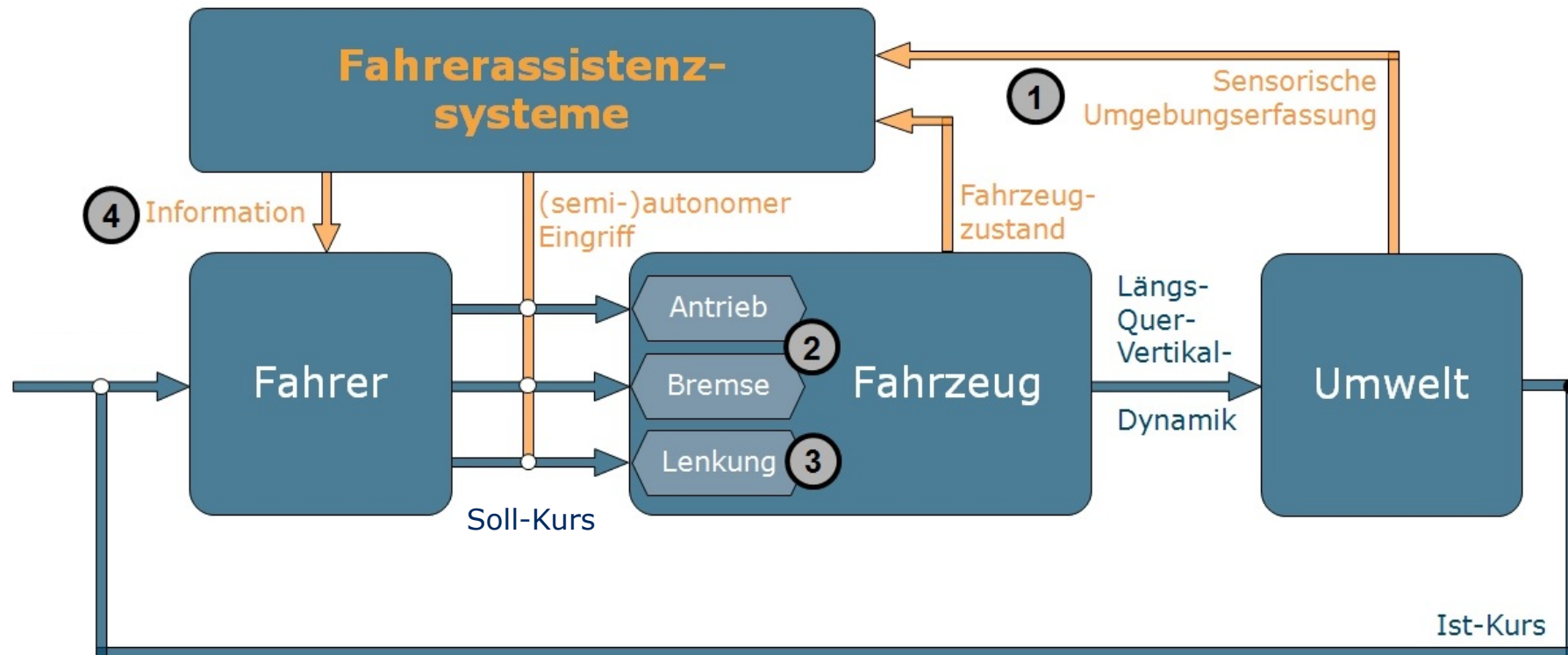
Modell eines Fahrerassistenzsystems



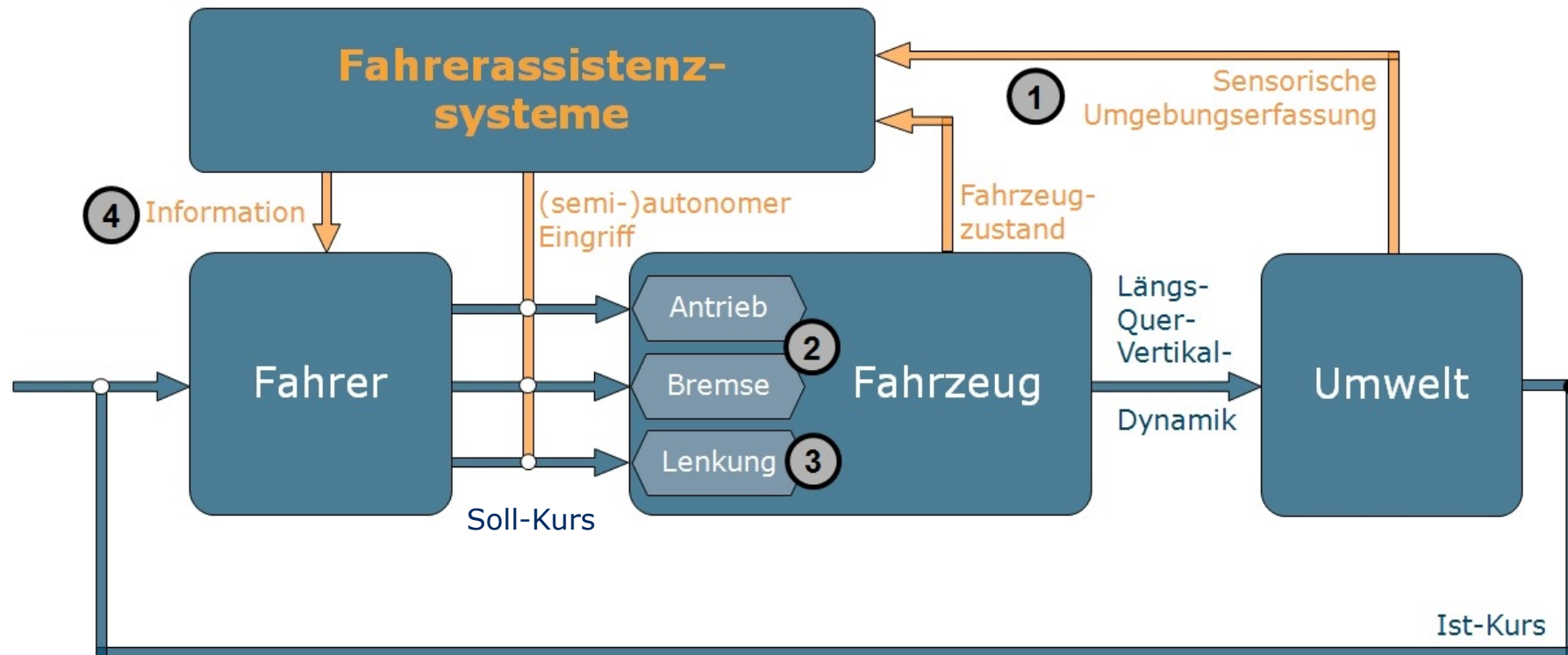
Modell eines Fahrerassistenzsystems



Modell eines Fahrerassistenzsystems

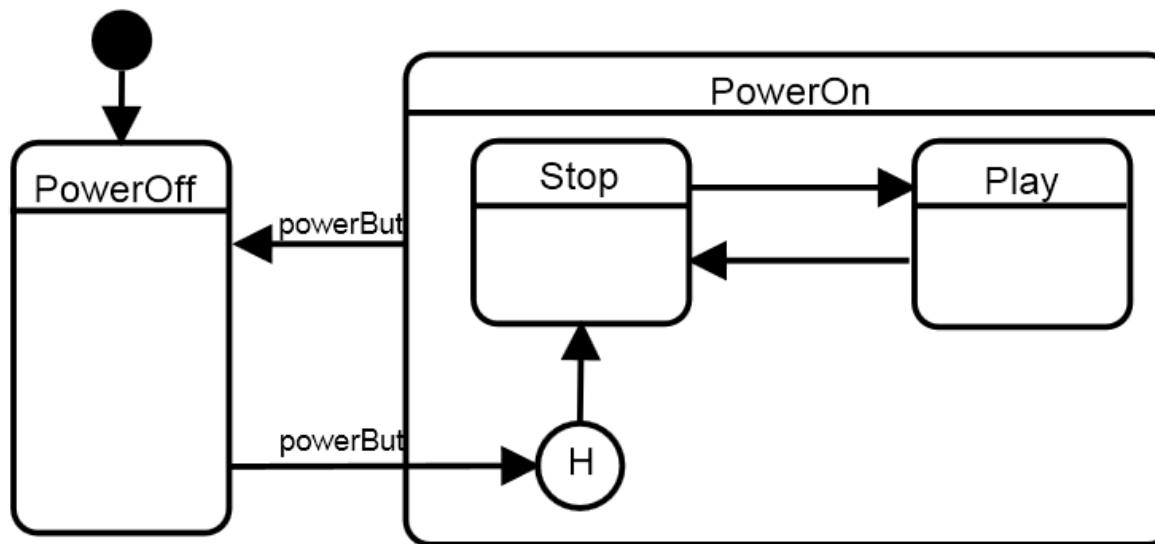


Modell eines Fahrerassistenzsystems



- Quelle: Prof. Bäker, TU Dresden, Lehrstuhl Fahrzeugmechatronik
http://tu-dresden.de/die_tu_dresden/fakultaeten/vkw/iad/professuren/fm/forschung/schwpkt/assistenz

Vergleich Modell - Code



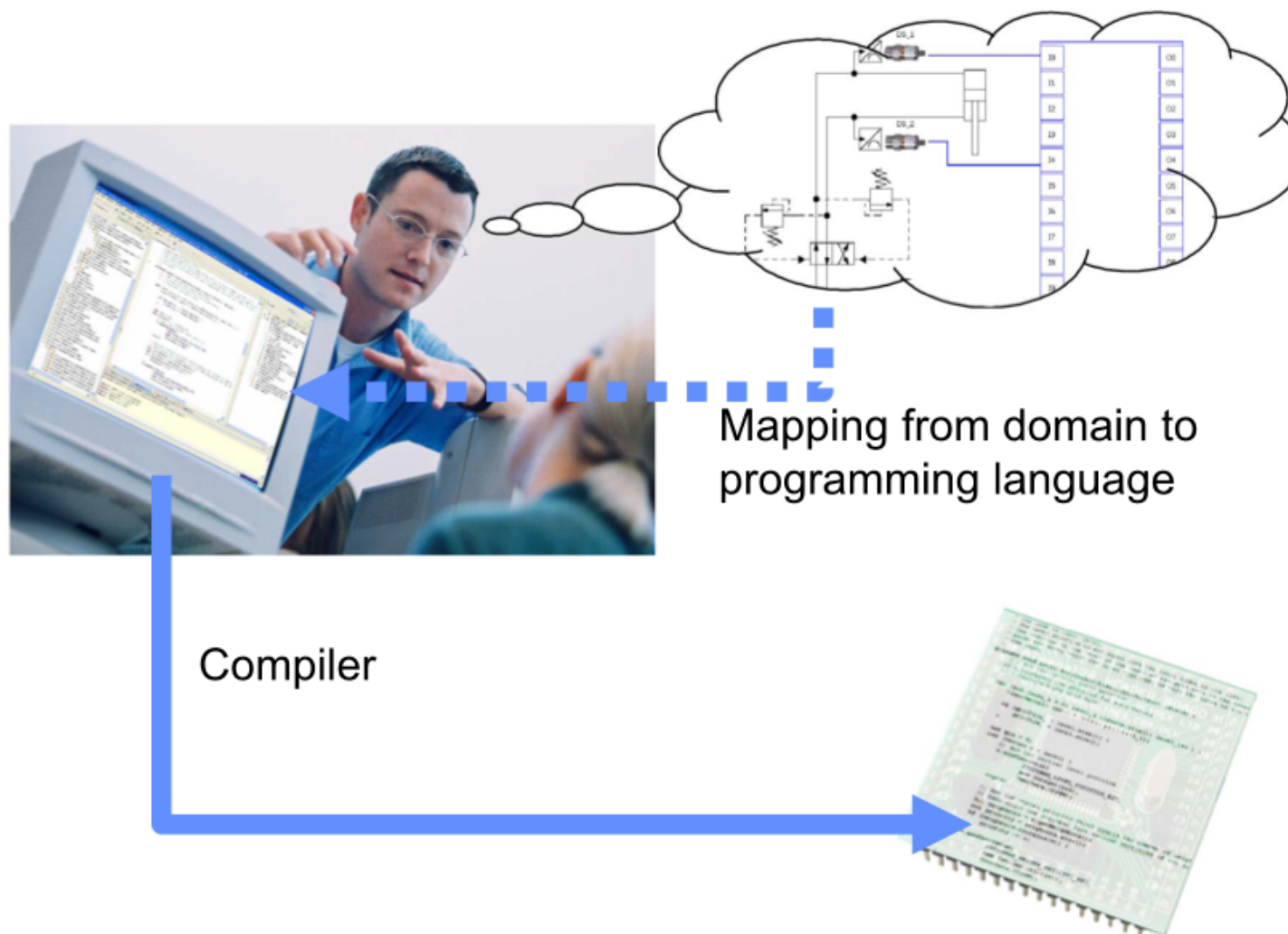
a) state chart

```

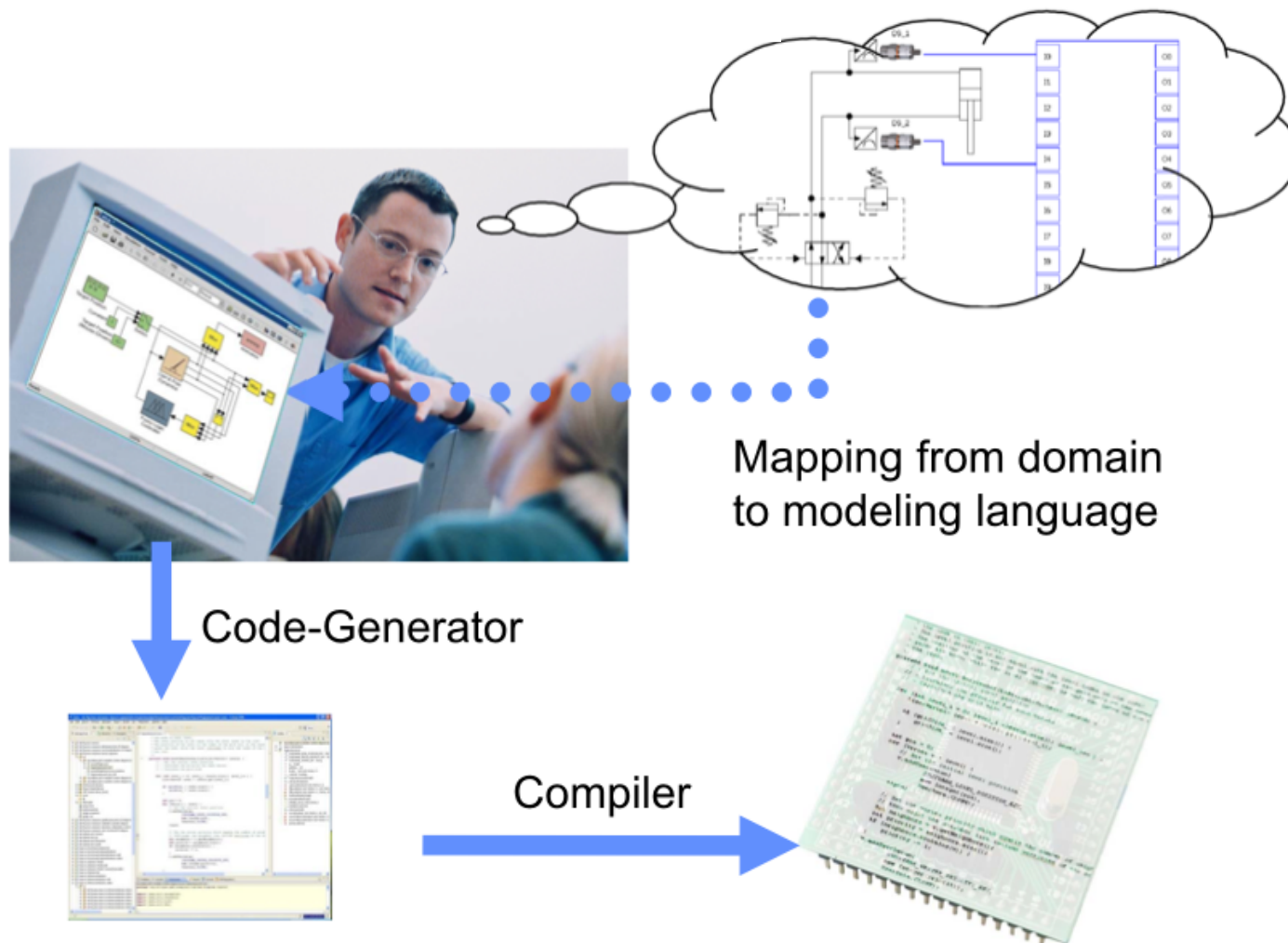
class Stereo {
    state on {
        activate() {
            // ... 
        }
    }
    state off {
        activate() {
            // ... 
        }
    }
    state stop {
        activate() {
            // ... 
        }
    }
    state play {
        activate() {
            // ... 
        }
    }
    state h {
        activate() {
            // ... 
        }
    }
    // ... 
}
  
```

b) generated Java method headers

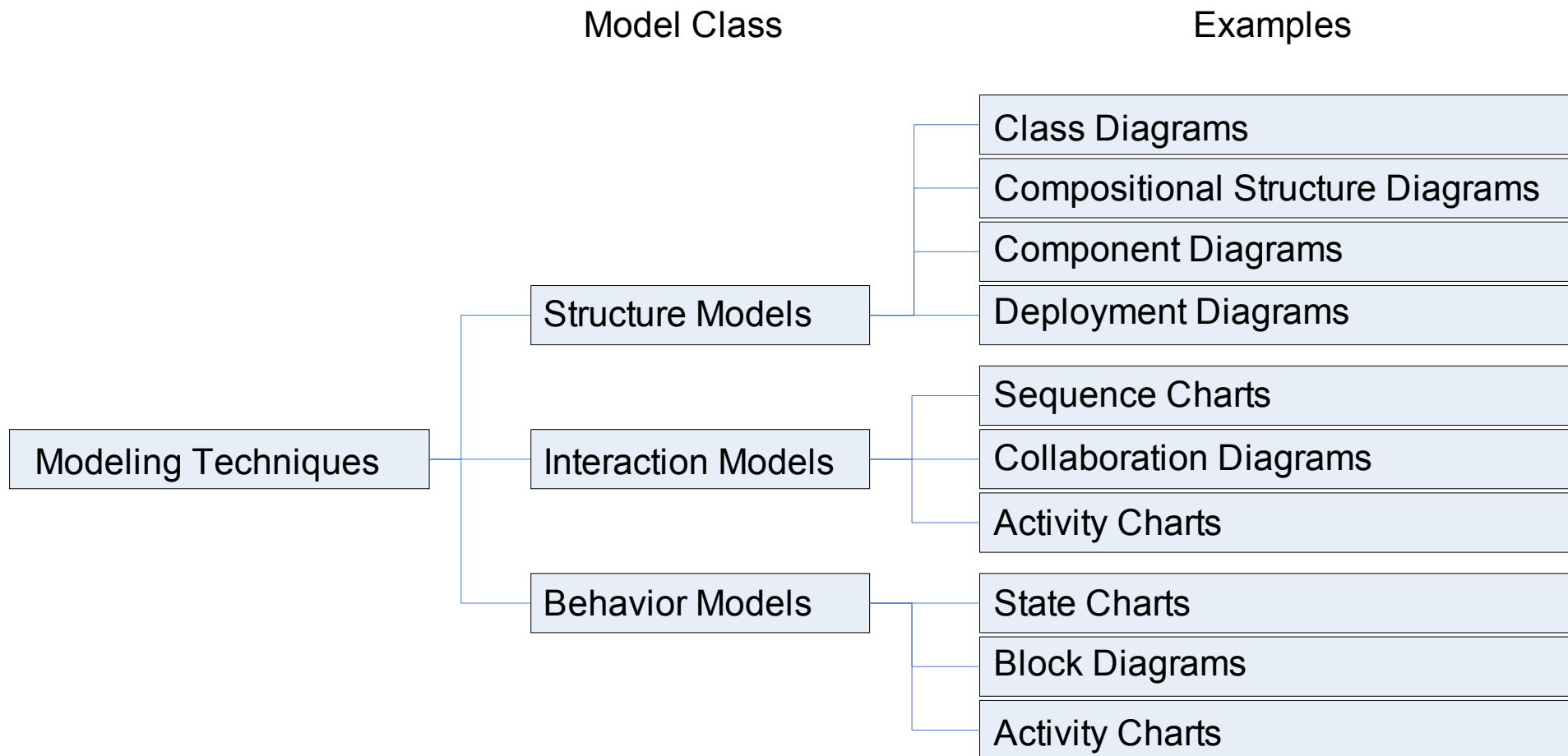
Coding



Modellbasierte Entwicklung



Klassifikation der Modellierungstechniken



Modellklassen

- Strukturmodelle (Structure Models)
 - Definieren, aus welchen strukturierten Elementen das System besteht
 - Definieren, wie diese Elemente statisch verbunden sind
 - Beispiel: UML-Kompositionsstrukturdiagramm
- Interaktionsmodelle (Interaction Models)
 - Definieren, wie die Elemente dynamisch untereinander interagieren
 - Basieren oft auf „Szenarien“
 - Beispiel: Sequenzdiagramme
- Verhaltensmodelle (Behavior Models)
 - Definieren das Verhalten der Systemelemente
 - Verhaltensmodelle sollten vollständig sein, haben eine klare Semantik und sind ausreichend formal, um Code zu generieren
 - Beispiel: Zustandsdiagramme

Weitere Informationen (1)

- Werkzeuge für MBD (Auswahl)
 - Rational Rhapsody (IBM)
 - Simulink (Mathworks) Die beiden wichtigsten Modellierungswerkzeuge
 - ASCET (ETAS) für Automotive
- MES Model Engineering Solutions www.model-engineers.com
- Gastvorträge in der Vorlesung Automotive Software Engineering
 - Jens O. Schindler, ITI GmbH, Dresden (Sommersemester 2014)
Modellbasierter Entwurf und Simulation physikalisch-technischer Systeme, 0D bis 3D Modellierung und Integration in den System Engineering Prozess
 - Dr.-Ing. Rocco Deutschmann, TraceTronic GmbH, Dresden (Sommersemester 2014)
Test, Simulation und Verifikation
 - Achim Rössler, Vector Informatik GmbH, Stuttgart (Sommersemester 2013)
Systemarchitektur und Systemoptimierung in der Automobilentwicklung: Das Werkzeug PREEvision

Weitere Informationen (2)

- Literaturhinweise:
 - P. Liggesmeyer, D. Rombach (Hrsg.): Software Engineering eingebetteter Systeme, Elsevier, 2005.
 - K. Berns, B. Schürmann, M. Trapp: Eingebettete Systeme, Springer, 2010.

Funktionsmodellierung (1)

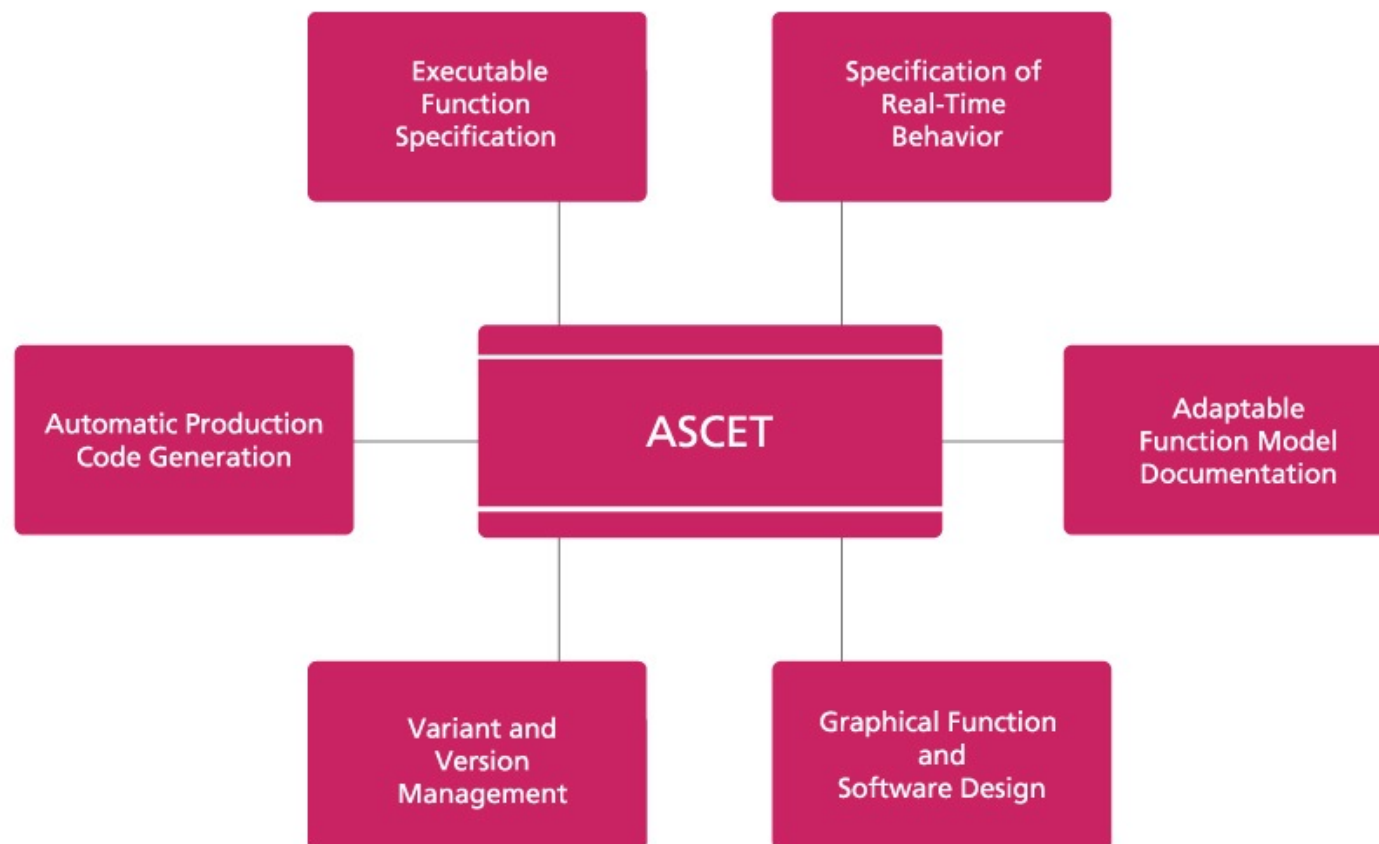
- Alle Antriebsfunktionen wurden mit dem Werkzeug ASCET entwickelt. Hierbei kamen Blockdiagramme, Zustandsautomaten und auch bedingte Tabellen zum Einsatz.
- http://www.etas.com/de/products/ascet_software_products.php
- Die ASCET Produktfamilie ermöglicht die modellbasierten Entwicklung von Anwendungssoftware (engl. Application SW) und die automatische Codegenerierung aus diesen Modellen. ASCET wurde speziell entwickelt, um den spezifischen Automotive-Anforderungen an komplexe Embedded Software mit Echtzeit-, Effizienz- und Sicherheitsrestriktionen gerecht zu werden.
- Vom ersten Komponentenentwurf mittels Blockdiagrammen und Zustandsautomaten bis zur automatischen Generierung des Codes für das Mikrocontrollersystem fügen sich die ASCET-Komponenten nahtlos in die Prozesse ein.
- Seit 1997 wird ASCET erfolgreich eingesetzt, insbesondere auch für die Entwicklung sicherheitskritischer Software rund um Brems- und Lenksysteme sowie bei Motormanagementsystemen. Mit ASCET entwickelte Software ist seit dem in über 68 Millionen Steuergeräten zu finden. Diese langjährige Erfahrung garantiert eine effiziente und sichere Softwareentwicklung mit gleich bleibend hoher Qualität und ermöglicht es, Kosten-, Qualitäts- und Time-to-Market-Ziele sicher zu erreichen.

Funktionsmodellierung (2)

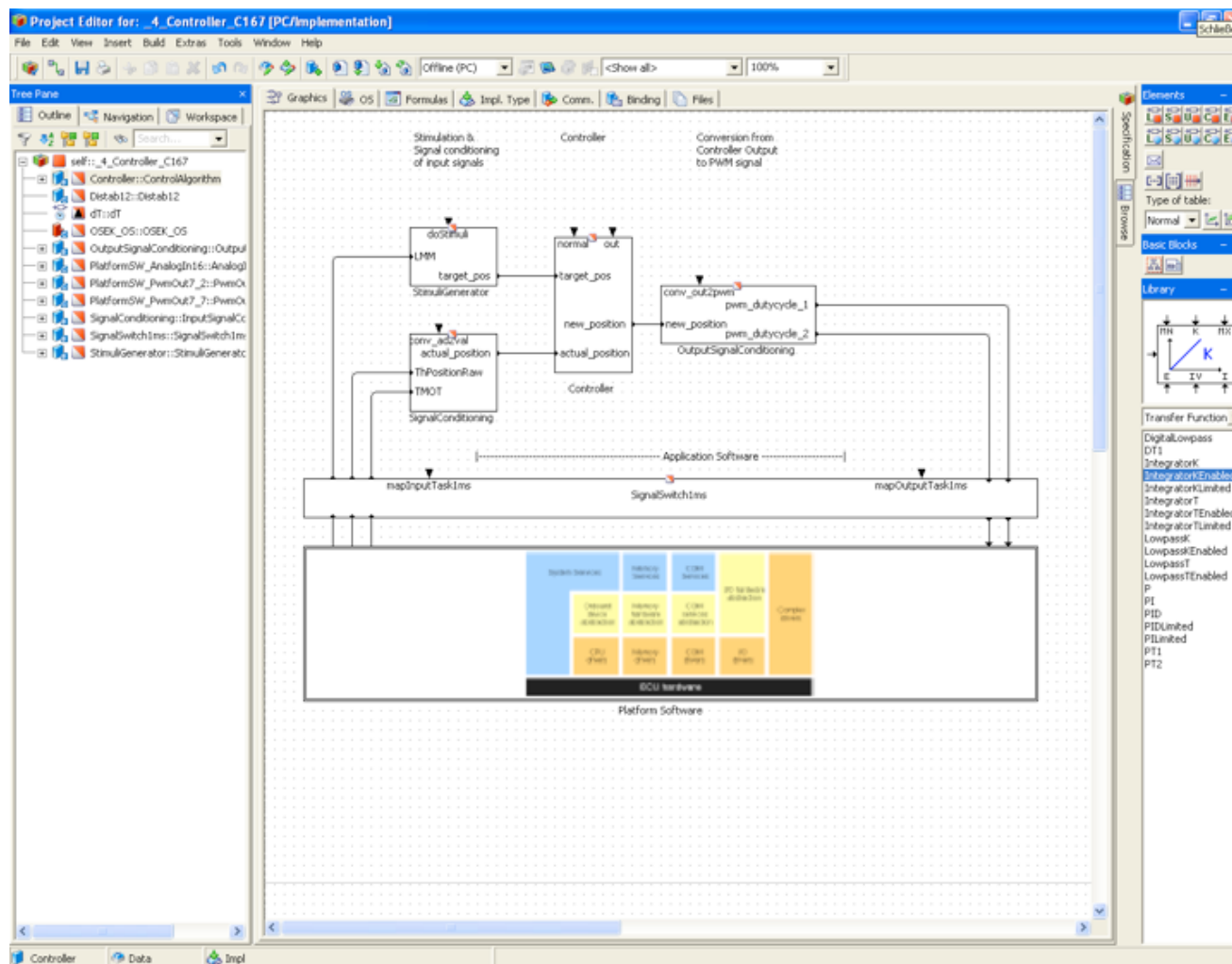
Funktionen auf einen Blick

- **Modellbasierte Entwicklung** von Automotive Software mit **Echtzeit-Anforderungen**
- Automatische und hoch effiziente **Generierung des Steuergeräte-Seriencodes**
- Migration und Entwicklung von **AUTOSAR** Softwarekomponenten
- Einfache Integration in bestehende **Entwicklungsumgebungen**
- Einfache Integration von bestehendem **C-Code** und **Library-Funktionen**
- **MISRA**-konformer und zertifizierbarer Codegenerator (IEC 61508 bzw. **ISO 26262**)
- Flexible, anpassbare **Dokumentation** der Funktions- und Softwaremodelle
- Import von **Simulink®**- und **UML**-Modellen
- Einfache Anbindung von Software-**Konfigurationsmanagement**-Systemen

Funktionsmodellierung (3)



Funktionsmodellierung (4)



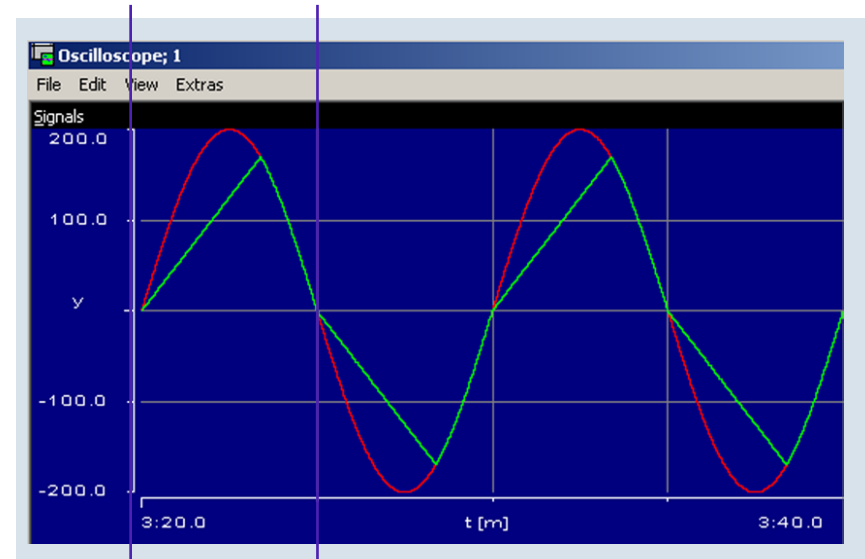
Entwicklung einer Antriebssteuerung für ein Hybridfahrzeug in einer Rapid Prototyping-Umgebung

1. Einleitung
2. Das Fahrzeug
3. Die Bordnetz-Architektur
4. Funktionsentwicklung mit ASCET
 1. Fahrpedal-Logik
 2. Berechnung des maximalen Drehmoments
 3. Drehmomentverteilung
 4. Drehmomentanstiegsbegrenzung
 5. Funktionsmodellierung
- 5. Die Erprobung**
 - 1. Offline-Simulation**
 2. Rapid Prototyping
6. Höhere Antriebsfunktion
7. Fazit
8. Abkürzungsverzeichnis

Quelle: http://www.etas.com/de/products/download_center.php?entrylist=13992

Offline-Simulation (1)

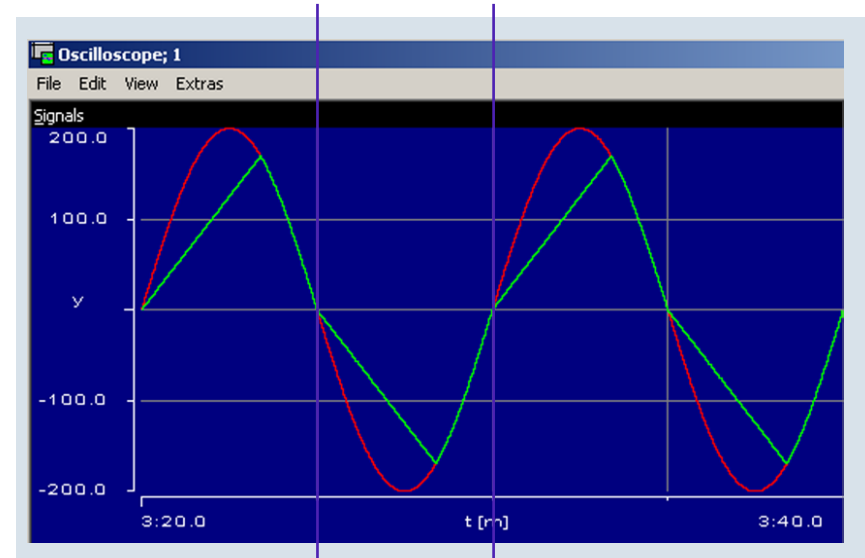
- Im ersten Schritt wurde das Verhalten der Softwarekomponenten durch Offline-Simulation erprobt. Die Abbildung zeigt ein Beispiel eines solchen Experiments, in dem die Drehmomentanstiegsbegrenzung getestet wird. In diesem sind der zeitliche Verlauf der Eingangsgröße (rote Kurve) und der Ausgangsgröße (grüne Kurve) dargestellt. Die Eingangsgröße wird hierbei durch eine Sinus-Funktion stimuliert. Wie man erkennen kann, baut sich das Drehmoment am Ausgang nur mit einer gewissen Steigung auf. In dem Moment, wo sich die grüne und die rote Kurve schneiden, nehmen beide Kurven (und damit die Eingangs- und die Ausgangsgröße) gemeinsam betragsmäßig ab.



Offline-Simulation der
Drehmomentanstiegsbegrenzung

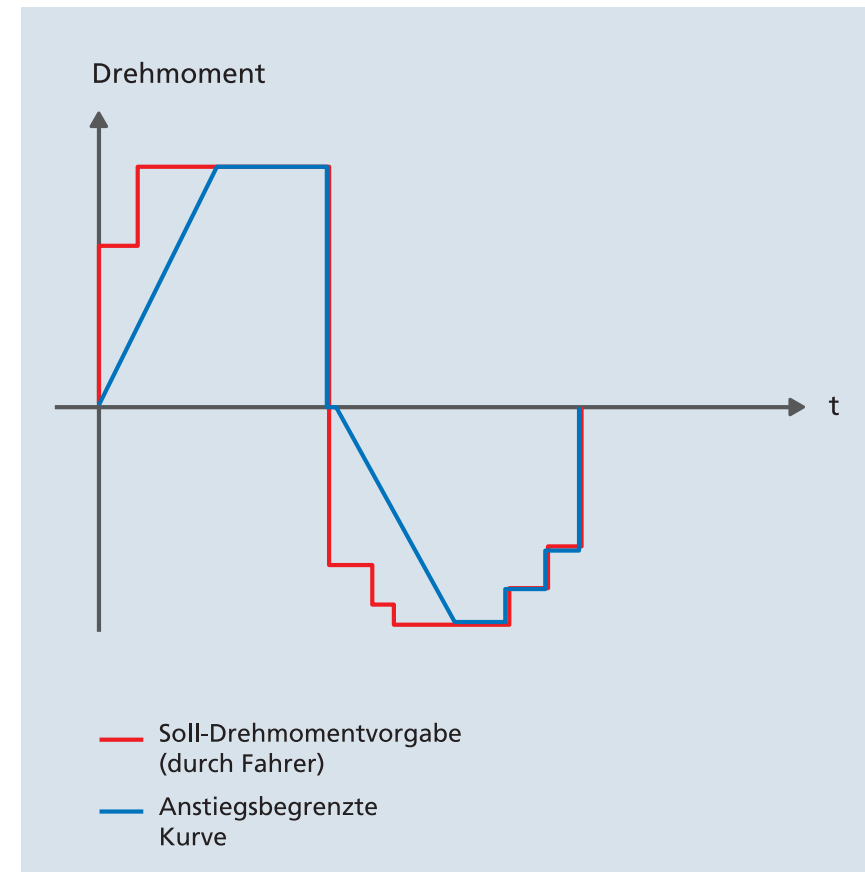
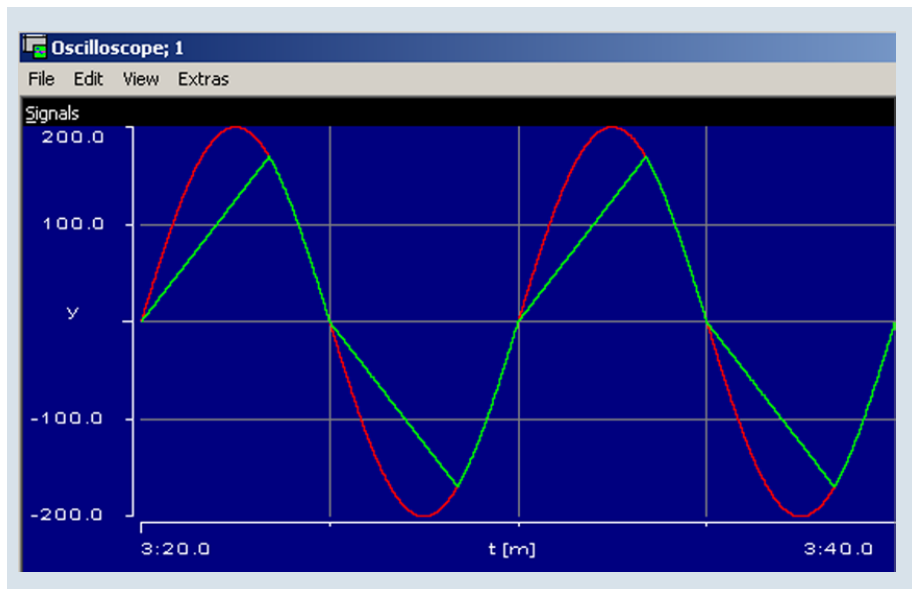
Offline-Simulation (2)

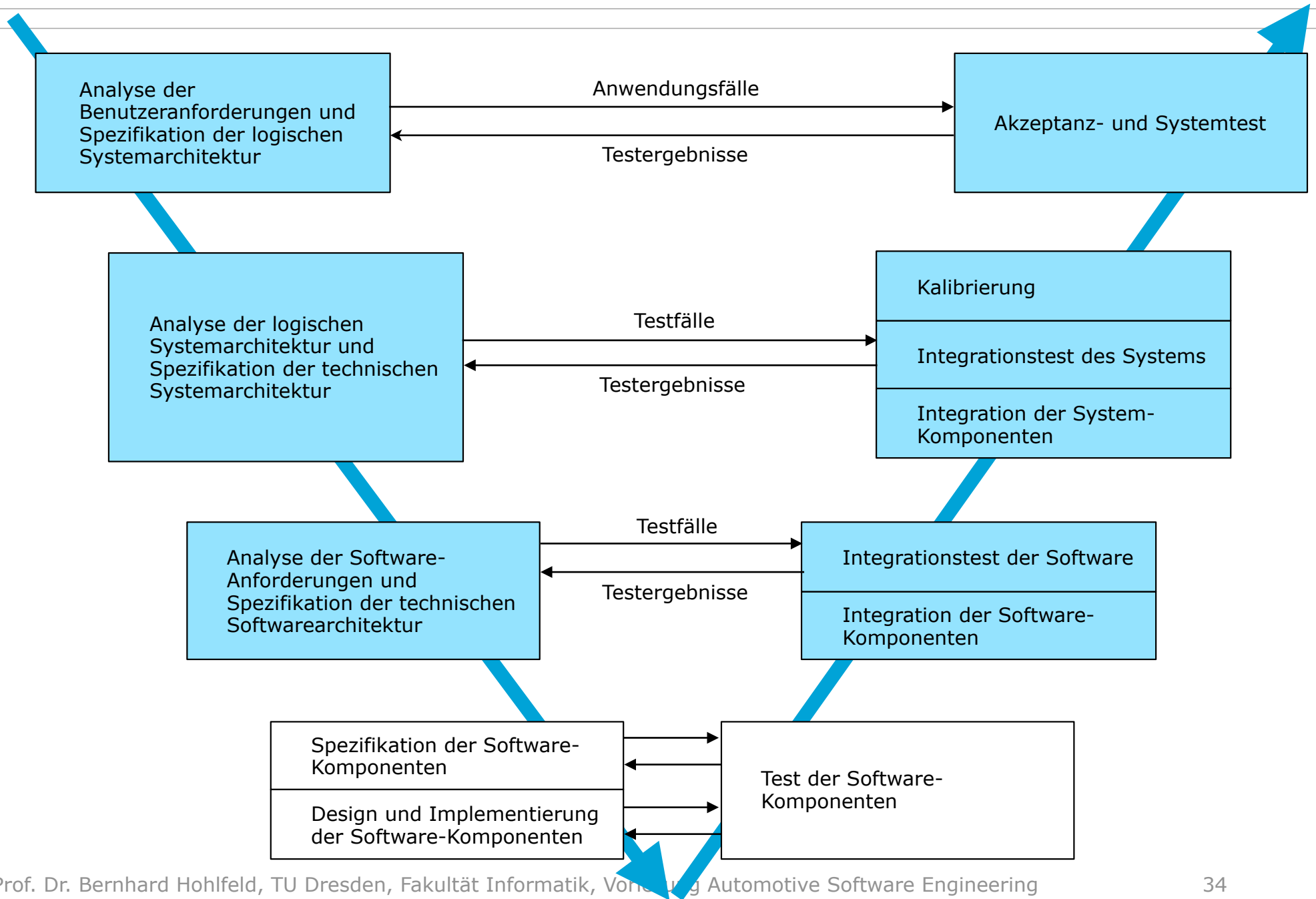
- Ist der Nullpunkt durchbrochen, baut sich das Drehmoment am Ausgang langsam mit der vorgegebenen Steigung in die andere Richtung auf. Somit kann man sich vom korrekten Verhalten der Klasse überzeugen.



Offline-Simulation der
Drehmomentanstiegsbegrenzung

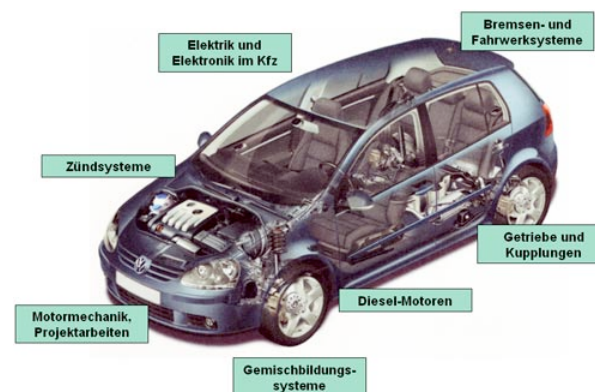
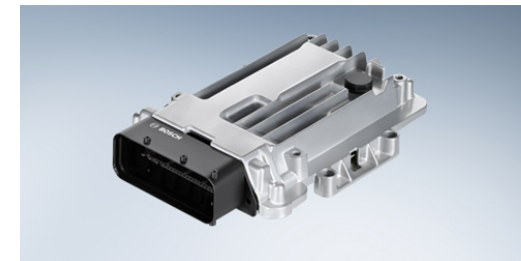
Offline-Simulation (3)





Begriffsdefinitionen

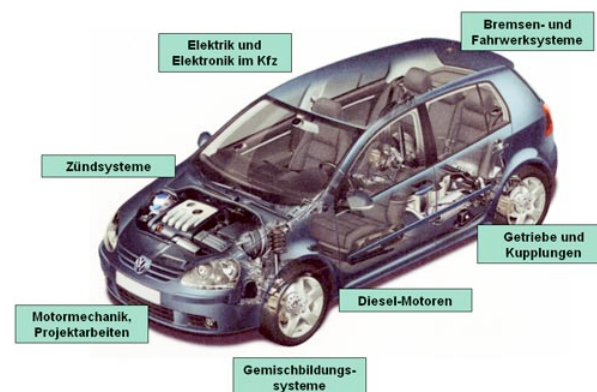
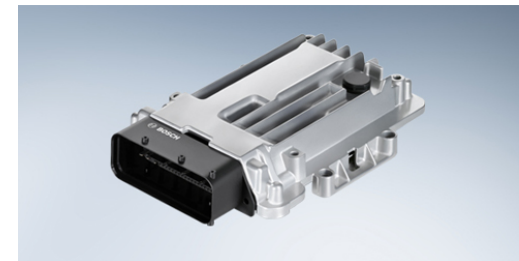
- MIL Model in the Loop, SIL Software in the Loop
 - Modell / SG-Software läuft auf simuliertem SG, das von simulierter Fahrzeugumgebung (rechnergenerierten Sensor- und Bussignalen) gespeist wird.
- HIL Hardware in the Loop
 - Reale SG-Hardware wird von simulierter Fahrzeugumgebung gespeist
- Prototyp
 - Simuliertes Steuergerät im Fahrzeug ("PC im Kofferraum")
- Prüfstand, Fahrversuch
 - Steuergerät im Fahrzeug unter Laborbedingungen bzw. auf der Strasse (Testgelände, öffentliches Strassennetz)



	Steuergerät	
Umgebung, Fahrzeug	simuliert	real
simuliert	MIL, SIL	HIL
real	Prototyp	Prüfstand, Fahrversuch

Begriffsdefinitionen

- MIL Model in the Loop, SIL Software in the Loop
 - Modell / SG-Software läuft auf simuliertem SG, das von simulierter Fahrzeugumgebung (rechnergenerierten Sensor- und Bussignalen) gespeist wird.
- HIL Hardware in the Loop
 - Reale SG-Hardware wird von simulierter Fahrzeugumgebung gespeist
- Prototyp
 - Simuliertes Steuergerät im Fahrzeug ("PC im Kofferraum")
- Prüfstand, Fahrversuch
 - Steuergerät im Fahrzeug unter Laborbedingungen bzw. auf der Strasse (Testgelände, öffentliches Strassennetz)



	Steuergerät	
Umgebung, Fahrzeug	simuliert	real
simuliert	MIL, SIL	HIL
real	Prototyp	Prüfstand, Fahrversuch

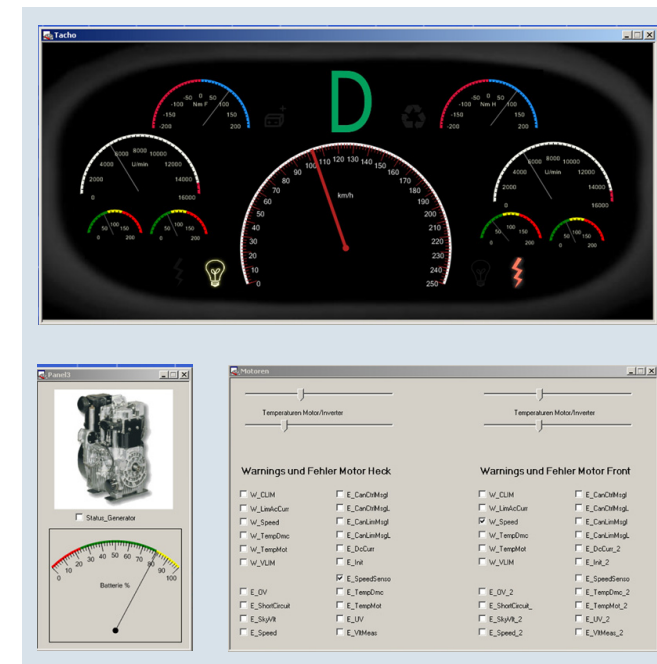
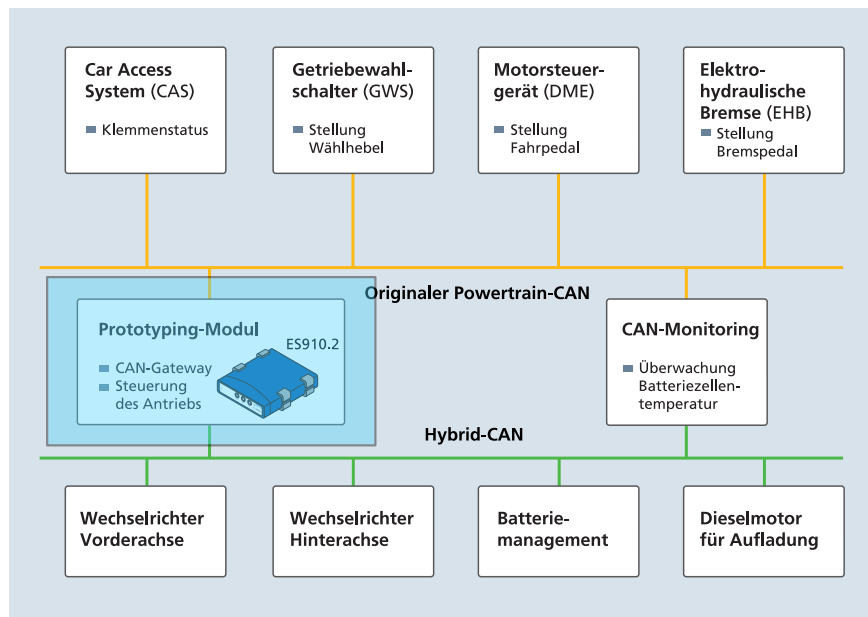
Entwicklung einer Antriebssteuerung für ein Hybridfahrzeug in einer Rapid Prototyping-Umgebung

1. Einleitung
2. Das Fahrzeug
3. Die Bordnetz-Architektur
4. Funktionsentwicklung mit ASCET
 1. Fahrpedal-Logik
 2. Berechnung des maximalen Drehmoments
 3. Drehmomentverteilung
 4. Drehmomentanstiegsbegrenzung
 5. Funktionsmodellierung
- 5. Die Erprobung**
 1. Offline-Simulation
 - 2. Rapid Prototyping**
6. Höhere Antriebsfunktion
7. Fazit
8. Abkürzungsverzeichnis

Quelle: http://www.etas.com/de/products/download_center.php?entrylist=13992

Rapid Prototyping (1)

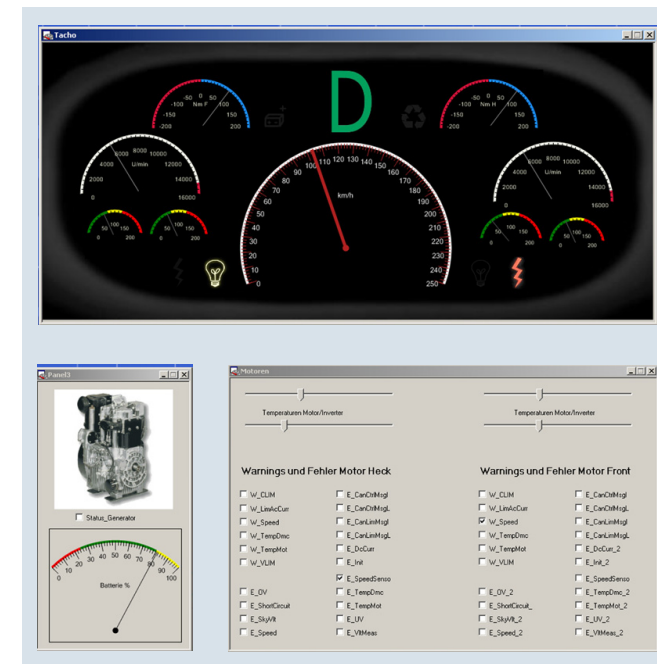
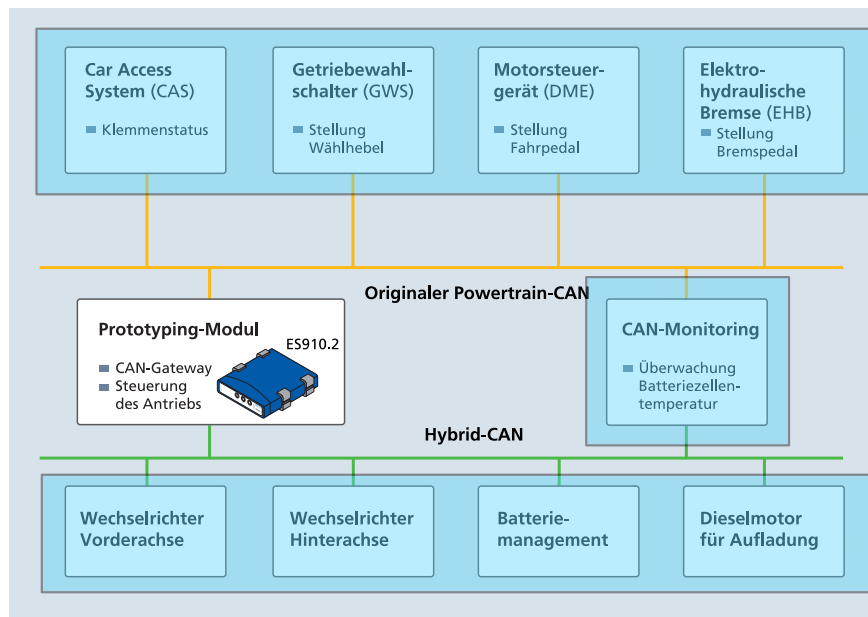
- Im nächsten Schritt wurde das Funktionsmodell mit Hilfe des Werkzeugs INTECRIO auf das **Rapid Prototyping-Modul ES910** gebracht. Bevor das Modul im realen Fahrzeug getestet wurde, erfolgte ein weiterer Absicherungsschritt der Gesamtfunktionalität gegenüber einer Restbussimulation (siehe Abbildung).



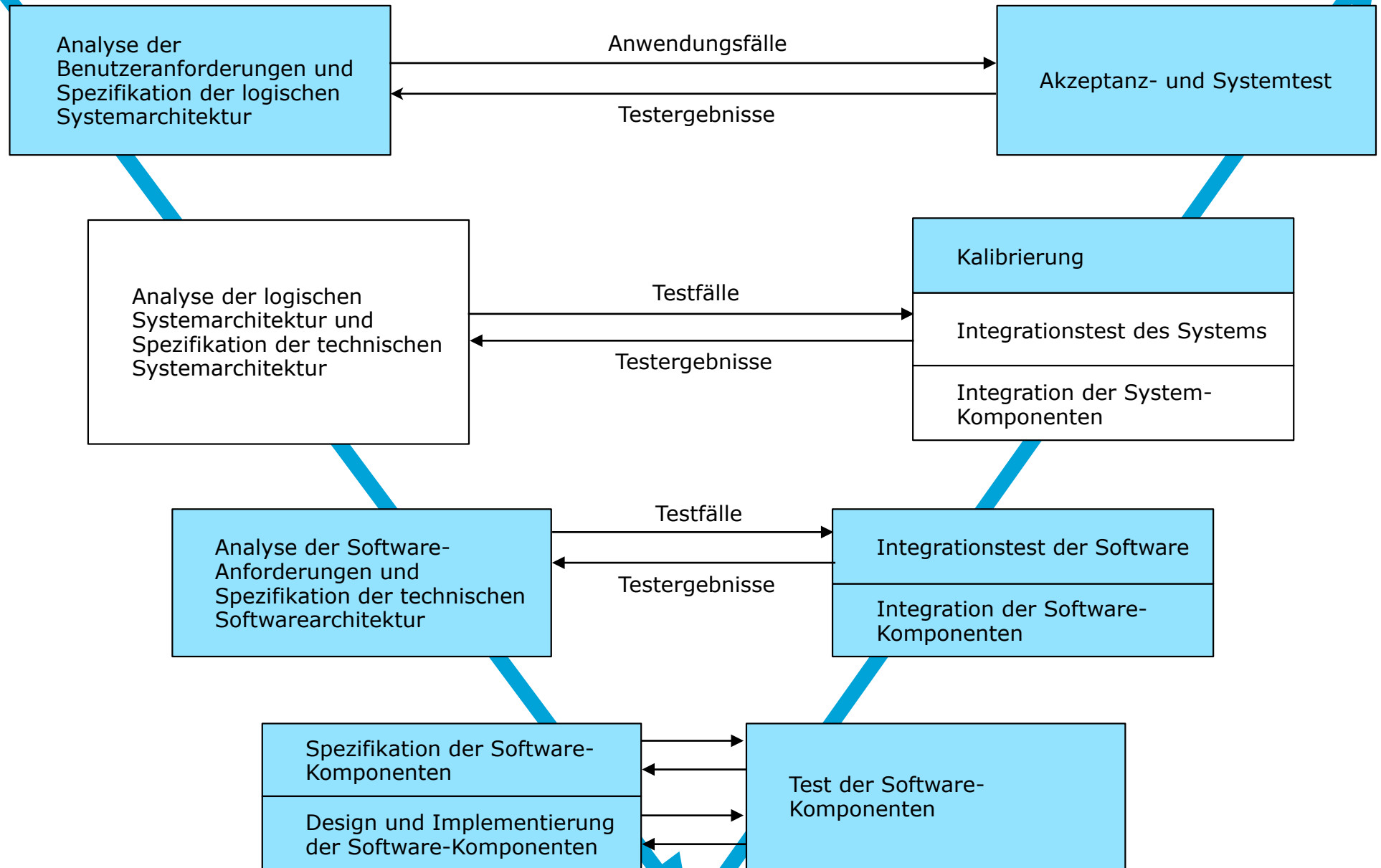
Panels der Restbussimulation

Rapid Prototyping (2)

- Im nächsten Schritt wurde das Funktionsmodell mit Hilfe des Werkzeugs INTECRIO auf das Rapid Prototyping-Modul ES910 gebracht. Bevor das Modul im realen Fahrzeug getestet wurde, erfolgte ein weiterer Absicherungsschritt der Gesamtfunktionalität gegenüber einer **Restbussimulation** (siehe Abbildung).

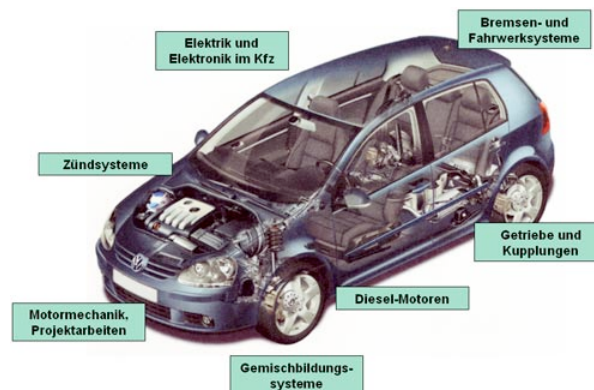
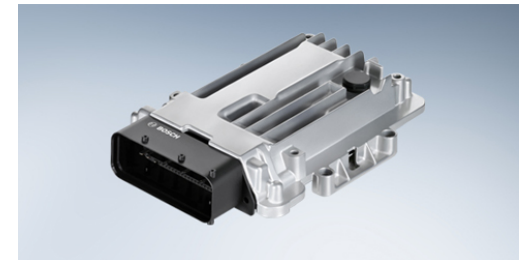


Panels der **Restbussimulation**



Begriffsdefinitionen

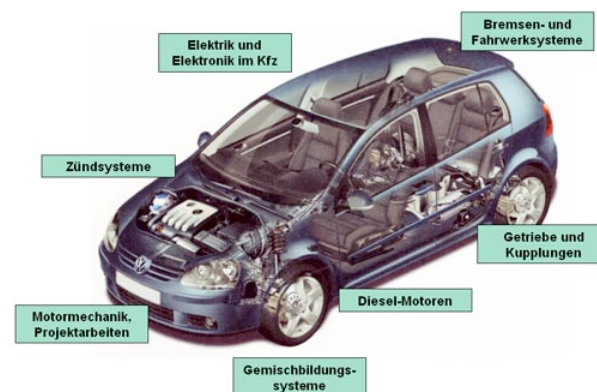
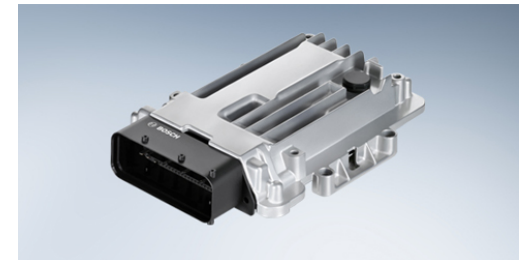
- MIL Model in the Loop, SIL Software in the Loop
 - Modell / SG-Software läuft auf simuliertem SG, das von simulierter Fahrzeugumgebung (rechnergenerierten Sensor- und Bussignalen) gespeist wird.
- HIL Hardware in the Loop
 - Reale SG-Hardware wird von simulierter Fahrzeugumgebung gespeist
- Prototyp
 - Simuliertes Steuergerät im Fahrzeug ("PC im Kofferraum")
- Prüfstand, Fahrversuch
 - Steuergerät im Fahrzeug unter Laborbedingungen bzw. auf der Strasse (Testgelände, öffentliches Strassennetz)



	Steuergerät	
Umgebung, Fahrzeug	simuliert	real
simuliert	MIL, SIL	HIL
real	Prototyp	Prüfstand, Fahrversuch

Begriffsdefinitionen

- MIL Model in the Loop, SIL Software in the Loop
 - Modell / SG-Software läuft auf simuliertem SG, das von simulierter Fahrzeugumgebung (rechnergenerierten Sensor- und Bussignalen) gespeist wird.
- HIL Hardware in the Loop
 - Reale SG-Hardware wird von simulierter Fahrzeugumgebung gespeist
- Prototyp
 - Simuliertes Steuergerät im Fahrzeug ("PC im Kofferraum")
- Prüfstand, Fahrversuch
 - Steuergerät im Fahrzeug unter Laborbedingungen bzw. auf der Strasse (Testgelände, öffentliches Strassennetz)

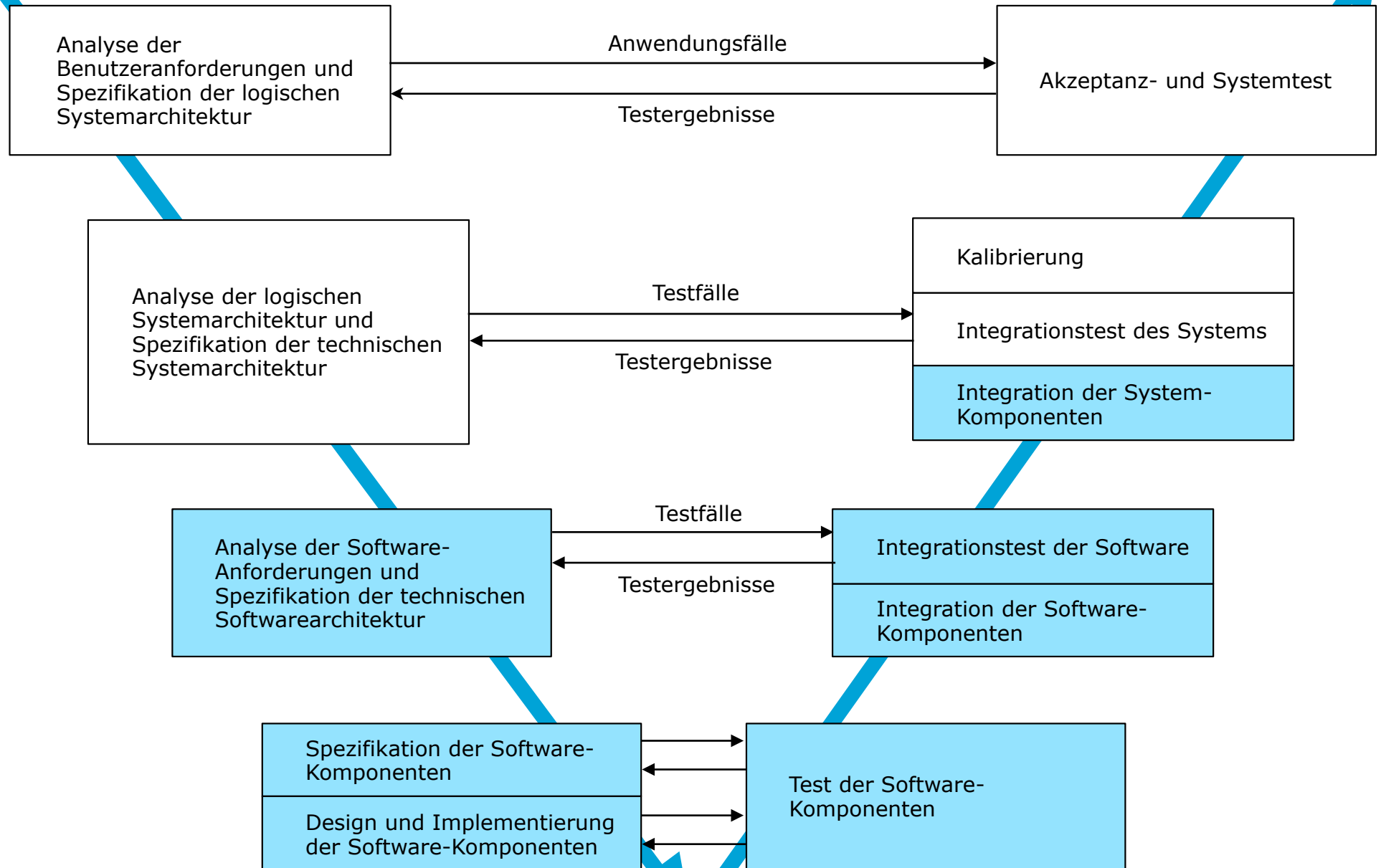


	Steuergerät	
Umgebung, Fahrzeug	simuliert	real
simuliert	MIL, SIL	HIL
real	Prototyp	Prüfstand, Fahrversuch

Rapid Prototyping (3)

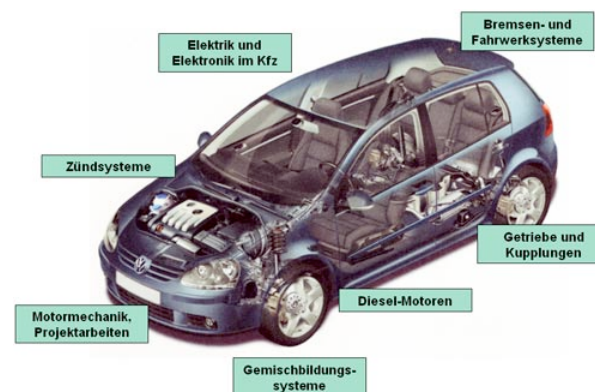
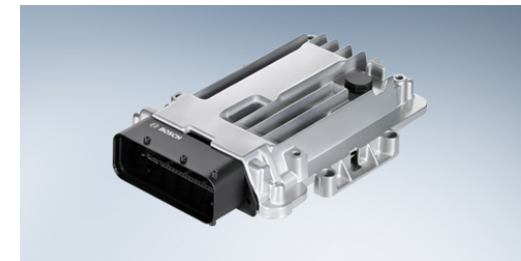
- Derzeit erfolgt die Erprobung des neuen Antriebssystems sowohl auf einem Rollenprüfstand als auch auf der Straße. Ziel der Erprobung ist es, durch Veränderung vielfältiger Parametersätze das Fahrverhalten zu optimieren. Im Zentrum steht dabei die Drehmomentverteilung auf die beiden Achsen des Fahrzeugs unter Berücksichtigung der vorhandenen Energie, der Temperatur aller Komponenten und der Fahrzeuggeschwindigkeit. Ferner wird mit unterschiedlichen Fahrpedalkennlinien experimentiert. Bei den ersten Versuchen hat sich gezeigt, dass man sich bei entsprechender Kalibrierung sehr schnell an das kombinierte Fahr-/Bremspedal gewöhnen kann.





Begriffsdefinitionen

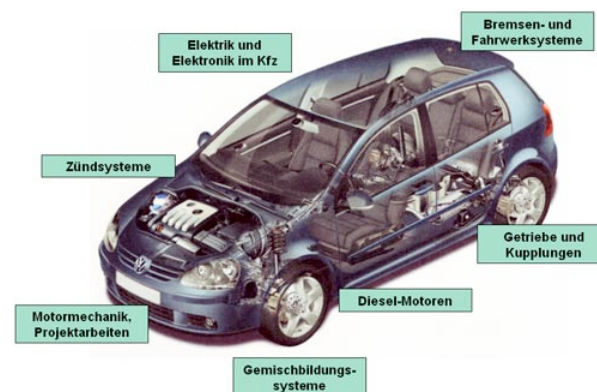
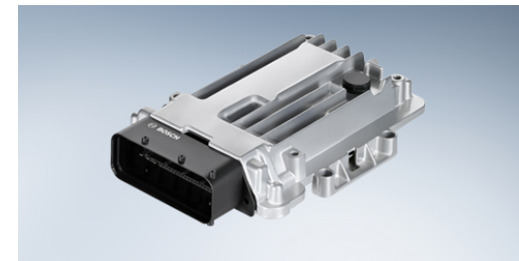
- MIL Model in the Loop, SIL Software in the Loop
 - Modell / SG-Software läuft auf simuliertem SG, das von simulierter Fahrzeugumgebung (rechnergenerierten Sensor- und Bussignalen) gespeist wird.
- HIL Hardware in the Loop
 - Reale SG-Hardware wird von simulierter Fahrzeugumgebung gespeist
- Prototyp
 - Simuliertes Steuergerät im Fahrzeug ("PC im Kofferraum")
- Prüfstand, Fahrversuch
 - Steuergerät im Fahrzeug unter Laborbedingungen bzw. auf der Strasse (Testgelände, öffentliches Strassennetz)



	Steuergerät	
Umgebung, Fahrzeug	simuliert	real
simuliert	MIL, SIL	HIL
real	Prototyp	Prüfstand, Fahrversuch

Begriffsdefinitionen

- MIL Model in the Loop, SIL Software in the Loop
 - Modell / SG-Software läuft auf simuliertem SG, das von simulierter Fahrzeugumgebung (rechnergenerierten Sensor- und Bussignalen) gespeist wird.
- HIL Hardware in the Loop
 - Reale SG-Hardware wird von simulierter Fahrzeugumgebung gespeist
- Prototyp
 - Simuliertes Steuergerät im Fahrzeug ("PC im Kofferraum")
- Prüfstand, Fahrversuch
 - Steuergerät im Fahrzeug unter Laborbedingungen bzw. auf der Strasse (Testgelände, öffentliches Strassennetz)



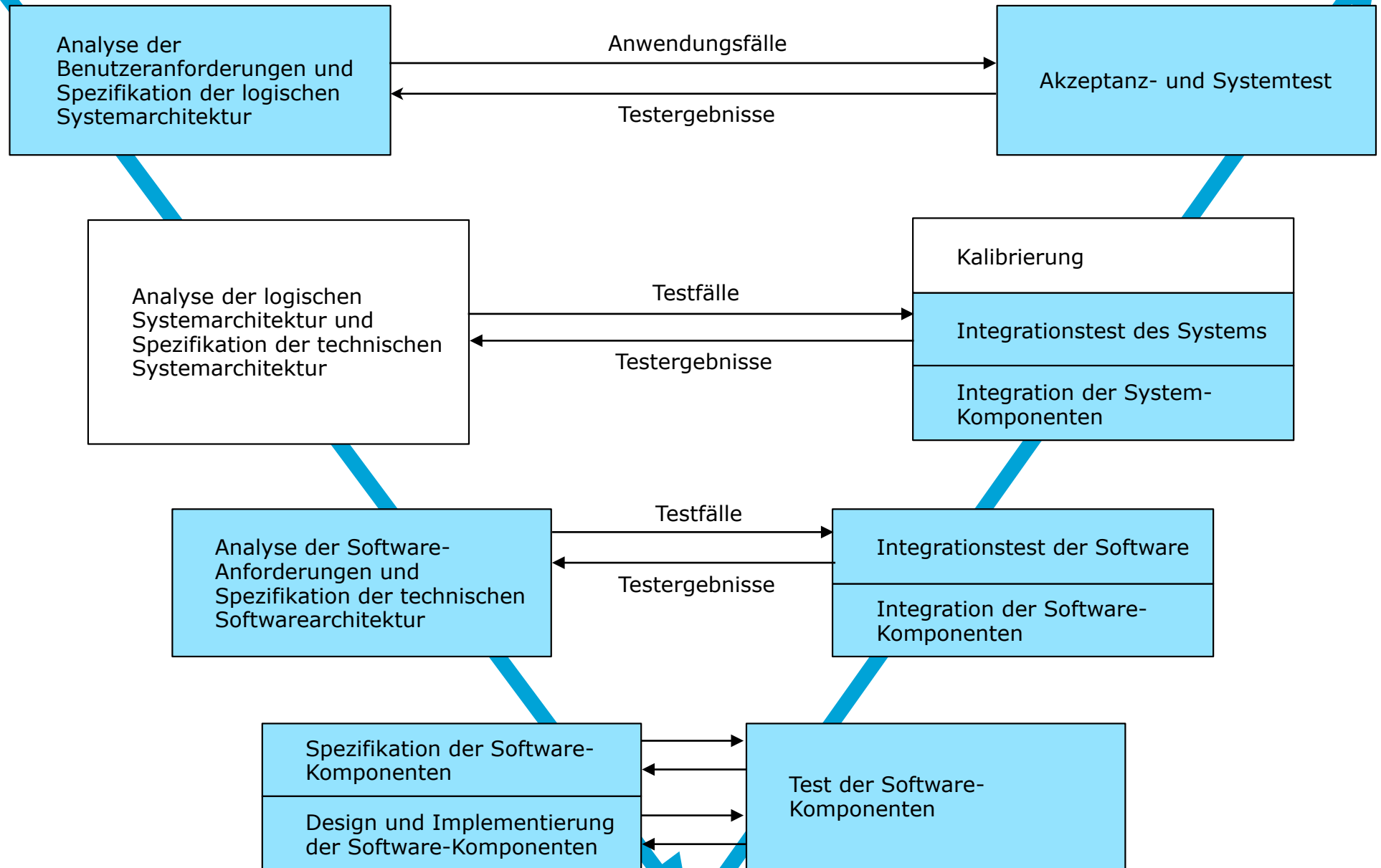
	Steuergerät	
Umgebung, Fahrzeug	simuliert	real
simuliert	MIL, SIL	HIL
real	Prototyp	Prüfstand, Fahrversuch

Rapid Prototyping (4)

- Derzeit werden die Erprobungen noch direkt über das Werkzeug INTECRIO gemacht (siehe Abbildung). Maschinenbau- und Informatikstudenten arbeiten dabei Hand in Hand. Als nächster Schritt wird nun eine INCA-Oberfläche entwickelt, mit der die Maschinenbauer die Fahrfunktionen selbst kalibrieren können.



INTECRIO-Oberfläche zur Überwachung des Systems

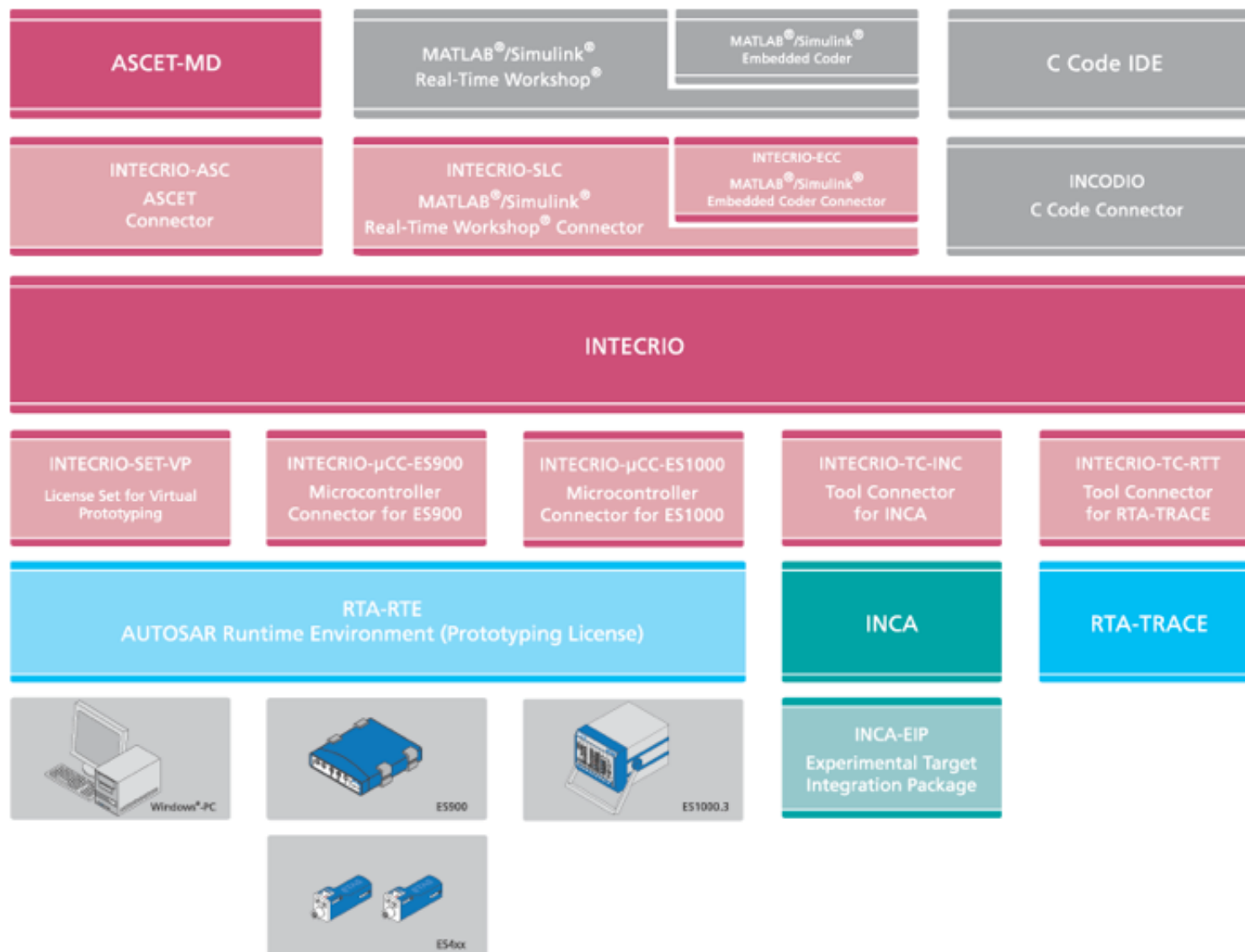


Rapid Prototyping (5)

- <http://www.etas.com/de/products/intecrio.php>
- INTECRIO ist ein ETAS-Softwareprodukt, welches das Prototyping elektronischer Systeme im Fahrzeug auf der Basis von **MATLAB®/Simulink®, ASCET-Modellen, C-Code-Modulen** und **AUTOSAR-Softwarekomponenten** ermöglicht. Es dient der Funktionsentwicklung als Integrations- und Experimentierplattform.
- Für die Verifikation und Validation im Fahrzeug unterstützt INTECRIO die modulare ES1000 VME Hardware-Serie von ETAS, die kompakten Rapid Prototyping Module ES900, die Mikromessmodule ES400 und ETK.
- Zur **Funktionsvalidation** bietet INTECRIO die Möglichkeit zum virtuellen **Prototyping** auf Windows® PCs direkt am Schreibtisch des Funktionsentwicklers und ermöglicht so die bereichsübergreifende Zusammenarbeit zwischen den Funktionsentwicklern auf der einen Seite und den Systementwicklern und Simulationsexperten auf der anderen Seite. Auf diese Weise bietet das Produkt neue Möglichkeiten für die frühen Fahrzeugentwicklungsphasen, wie zum Beispiel die Vorkalibrierung im Labor.

Rapid Prototyping (6)

- Offene Umgebung für **Rapid Prototyping** von elektronischen Systemen im Fahrzeug
- Integration von **MATLAB®/Simulink®** und **ASCET-Modellen** und **C-Code-Modulen**
- Integration von **AUTOSAR-Softwarekomponenten** durch die Verwendung von **AUTOSAR RTE**
- Test neuer Funktionen unter virtuellen Bedingungen am PC (**virtuelles Prototyping**) oder unter realen Bedingungen (**Rapid Prototyping**)
- Unterstützung verschiedenster **Steuergeräte-, Bus-, Sensor- und Aktuatorschnittstellen** einschließlich ETK (Universelle Steuergeräte-Schnittstellen von ETAS, Abkürzung nicht erklärt)
- **Funktionstest** im Fahrzeug mit ETK-Bypass, dem modularen ES1000 Hardwaresystem oder dem ES910 Rapid Prototyping-Kompaktmodul in Verbindung mit den ES400 Mikromessmodulen
- **Offene Schnittstellen** für die Automatisierung und kundenspezifischen Anpassungen
- Intuitive grafische Benutzeroberfläche
- Nahtloses Zusammenspiel mit dem **Mess- und Kalibrierwerkzeug** INCA und dem Software Logic Analyzer RTA-TRACE



Entwicklung einer Antriebssteuerung für ein Hybridfahrzeug in einer Rapid Prototyping-Umgebung

1. Einleitung
2. Das Fahrzeug
3. Die Bordnetz-Architektur
4. Funktionsentwicklung mit ASCET
 1. Fahrpedal-Logik
 2. Berechnung des maximalen Drehmoments
 3. Drehmomentverteilung
 4. Drehmomentanstiegsbegrenzung
 5. Funktionsmodellierung
5. Die Erprobung
 1. Offline-Simulation
 2. Rapid Prototyping
- 6. Höhere Antriebsfunktion**
7. Fazit
8. Abkürzungsverzeichnis

Quelle: http://www.etas.com/de/products/download_center.php?entrylist=13992

Höhere Antriebsfunktion (1)

- Basierend auf den derzeitigen Basisfunktionen, entwickeln jetzt mehrere Studenten höhere Antriebsfunktionen. Dabei werden nicht nur **Komfortfunktionen**, wie z. B. **Tempomat** oder **Halten am Berg** entwickelt, sondern auch Sicherheitsfunktionen, wie z. B. eine **Schlupfregelung** für den elektrischen Antrieb. Auch diese Funktionen werden mit ASCET spezifiziert. Ziel ist es, einzelne gekapselte Softwarekomponenten nach **AUTOSAR** zu entwickeln. Im Vordergrund steht auch hier, wie beim gesamten Projekt, die Studenten **praxisnah** an **modernen Entwicklungsmethoden** auszubilden.



Überwachung der internen Betriebszustände mit INTECRIO

Höhere Antriebsfunktion (2)

- Es ist geplant, das Fahrzeug auch in den kommenden Semestern als Entwicklungs- und Ausbildungsplattform zu nutzen. Das Spektrum reicht dabei vom **4-Motorenantrieb** (bisher 2) über **Hochvoltsicherungen** bis hin zu **Batteriemanagementsystemen** sowie Anzeige- und Bedienkonzepten.



Serieller Plug-in-Hybrid auf
Basis eines BMW X5

Entwicklung einer Antriebssteuerung für ein Hybridfahrzeug in einer Rapid Prototyping-Umgebung

1. Einleitung
2. Das Fahrzeug
3. Die Bordnetz-Architektur
4. Funktionsentwicklung mit ASCET
 1. Fahrpedal-Logik
 2. Berechnung des maximalen Drehmoments
 3. Drehmomentverteilung
 4. Drehmomentanstiegsbegrenzung
 5. Funktionsmodellierung
5. Die Erprobung
 1. Offline-Simulation
 2. Rapid Prototyping
6. Höhere Antriebsfunktion
- 7. Fazit**
8. Abkürzungsverzeichnis

Quelle: http://www.etas.com/de/products/download_center.php?entrylist=13992

Fazit

- Die Hochschule Landshut hat mit dem Projekt X5 Hybrid bewiesen, dass es möglich ist, seriennahe Hybridfahrzeuge zu bauen, ohne auf den gewohnten Komfort eines konventionellen Fahrzeugs verzichten zu müssen.
- Für die Entwicklung der neuen Hybridfunktionen wurden die ETAS-Werkzeuge ASCET, INTCRIO, ES910 und INCA entlang des klassischen V-Modells eingesetzt. Damit konnten die Funktionen spezifiziert, abgesichert und letztendlich mit Rapid Prototyping-Technologie im Fahrzeug ausgeführt werden. So war es den Informatikern möglich, in relativ kurzer Zeit dem Fahrzeug der Maschinenbauer Leben einzuhauchen.
- Nähere Informationen zum Projekt finden Sie unter www.ex-drive.de.
- Der Beitrag ist erschienen bei: Sirch, Ottmar (Hrsg.): Elektrik/Elektronik in Hybrid- und Elektrofahrzeugen II „Entwicklung einer Antriebssteuerung für ein Hybridfahrzeug in einer Rapid-Prototyping-Umgebung“: Klaus Beck, Thomas Escherle, Dieter Nazareth, expert verlag, 2010.

Entwicklung einer Antriebssteuerung für ein Hybridfahrzeug in einer Rapid Prototyping-Umgebung

1. Einleitung
2. Das Fahrzeug
3. Die Bordnetz-Architektur
4. Funktionsentwicklung mit ASCET
 1. Fahrpedal-Logik
 2. Berechnung des maximalen Drehmoments
 3. Drehmomentverteilung
 4. Drehmomentanstiegsbegrenzung
 5. Funktionsmodellierung
5. Die Erprobung
 1. Offline-Simulation
 2. Rapid Prototyping
6. Höhere Antriebsfunktion
7. Fazit
- 8. Abkürzungsverzeichnis**

Quelle: http://www.etas.com/de/products/download_center.php?entrylist=13992

Abkürzungsverzeichnis

- CAN
Controller Area Network
Datenbus im Kraftfahrzeug und in der Automatisierungstechnik.
- CAS
Car Access System
Steuerung des Fahrzeugzugriffs, die u. a. den Status der Klemme 15 regelt.
- DMC
Digital Motion Control
Versorgt die Elektroantriebe mit Wechselspannung.
- DME
Digitale Motorelektrik Konventionelles Motorsteuergerät des BMW X5.
- EHB
Elektrohydraulische Bremse
Bremsanlage des BMW X5 mit Bremskraftverstärker, ABS und ESP.
- GWS
Getriebewahlschalter
Wählhebel des Automatikgetriebes.
- DSC
Dynamische Stabilitäts Control (sic!)
- PDC
Park Distance Control (BMW)
Parktronic (Mercedes)
- ACC
Adaptive Cruise Control (BMW)
DISTRONIC, Abstandsregeltempomat (Mercedes)

Offene Fragen

- „Basisfahrzeug war ein von der BMW AG zur Verfügung gestellter BMW X5. Aus diesem Fahrzeug wurde der gesamte konventionelle Antriebsstrang entfernt.“
- „Als Antrieb dienen nun zwei in den beiden Achsen verbaute Elektromotoren.“
- „Die dritte Sitzreihe des X5 musste den Lithium-Ionen-Batterien weichen, die zusammen eine Spannung von 400 V liefern.“



Offene Fragen

- „Basisfahrzeug war ein von der BMW AG zur Verfügung gestellter BMW X5. Aus diesem Fahrzeug wurde der gesamte konventionelle Antriebsstrang entfernt.“
- „Als Antrieb dienen nun zwei in den beiden Achsen verbaute Elektromotoren.“
- „Die dritte Sitzreihe des X5 musste den Lithium-Ionen-Batterien weichen, die zusammen eine Spannung von 400 V liefern.“



Offene Fragen

- „Basisfahrzeug war ein von der BMW AG zur Verfügung gestellter BMW X5. Aus diesem Fahrzeug wurde der gesamte konventionelle Antriebsstrang entfernt.“
- „Als Antrieb dienen nun zwei in den beiden Achsen verbaute Elektromotoren.“
- „Die dritte Sitzreihe des X5 musste den Lithium-Ionen-Batterien weichen, die zusammen eine Spannung von 400 V liefern.“



Offene Fragen

- „Basisfahrzeug war ein von der BMW AG zur Verfügung gestellter BMW X5. Aus diesem Fahrzeug wurde der gesamte konventionelle Antriebsstrang entfernt.“
- „Als Antrieb dienen nun zwei in den beiden Achsen verbaute Elektromotoren.“
- „Die dritte Sitzreihe des X5 musste den Lithium-Ionen-Batterien weichen, die zusammen eine Spannung von 400 V liefern.“



Offene Fragen

- „Basisfahrzeug war ein von der BMW AG zur Verfügung gestellter BMW X5. Aus diesem Fahrzeug wurde der gesamte konventionelle Antriebsstrang entfernt.“
- „Als Antrieb dienen nun zwei in den beiden Achsen verbaute Elektromotoren.“
- „Die dritte Sitzreihe des X5 musste den Lithium-Ionen-Batterien weichen, die zusammen eine Spannung von 400 V liefern.“



Offene Fragen

- „Basisfahrzeug war ein von der BMW AG zur Verfügung gestellter BMW X5. Aus diesem Fahrzeug wurde der gesamte konventionelle Antriebsstrang entfernt.“
- „Als Antrieb dienen nun zwei in den beiden Achsen verbaute Elektromotoren.“
- „Die dritte Sitzreihe des X5 musste den Lithium-Ionen-Batterien weichen, die zusammen eine Spannung von 400 V liefern.“

