

Systemtest und Systemsimulation

Team Testmethoden – Tools, Simulation

Volker Nieten
Dr. Christian Pigorsch
Thomas Würdinger
Dr. Paul Willutzki

BMW Group



Gliederung

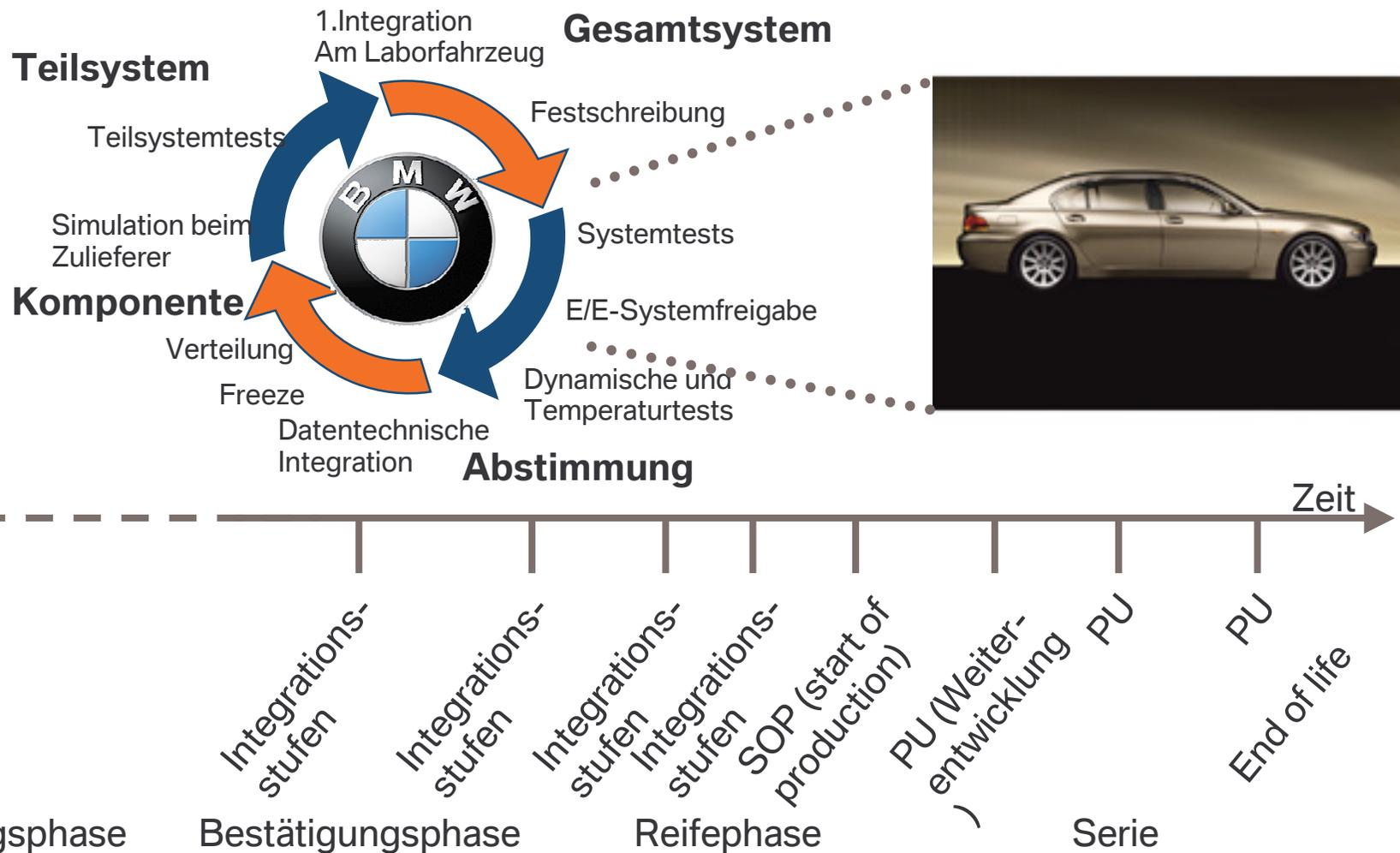
1. Entwicklungskette für die Systemintegration
2. Prozesskette für Buskommunikation
3. Restbussimulation
4. Busmonitoring
5. Testplattformen für das Gesamtsystem Elektrik /
Elektronik
6. Gesamtsystemtests und -methoden
7. Werkzeuge

Entwicklungskette Elektrik/Elektronik

Phasen innerhalb einer Integrationsstufe

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge



Entwicklungskette Elektrik/Elektronik

Phasen innerhalb einer Integrationsstufe

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge

	Ziel	Voraussetzungen	Ergebnisse	Tests und Testmethoden
Abstimmung	Stimmigkeit der Buskommunikation zwischen den Teilnehmern	Abstimmung aller beteiligten Partner	Bordnetz-Freeze, Prüfung von Daten-bankexport (SGML,XML)	Konsistenztests (Java), Bussimulationen und Busmonitoring, SIL
Komponente	Anlieferung einer Komponente in kompatibler und getesteter Ausführung an das Teilsystem	Detaillierte Spezifikationen, definierte HW- und SW-Standards	Validierte und verifizierte Komponenten	Funktionstests (System- und Kundenfunktionen), Softwaremetriken, Prüf- und Qualifikationsberichte, Restbussimulation, Busmonitoring, HiL (Hardware in the Loop Prüfstände), Testvekt.
Teilsystem	Integration von Steuergeräten einer Gruppe (z.B. Infotainment)	Alle Steuergeräte in entsprechender Qualität vorhanden	Validierte und verifizierte Teilsysteme	Standardisierte Prüfberichte, Checklisten, Restbussimulationen, Busmonitoring, HiL, weitere Testwerkzeuge
Gesamtsystem	Integration aller Komponenten in einem Fahrzeug	Validierte und verifizierte Teilsysteme	Fzg. Mit validierten System- und Kundenfunktionen, Robustheit gegen phys. und datentechnische Störungen	Fahrzeugchecklisten, Testfahrten, Busmonitoring, automatisierte Testwerkzeuge

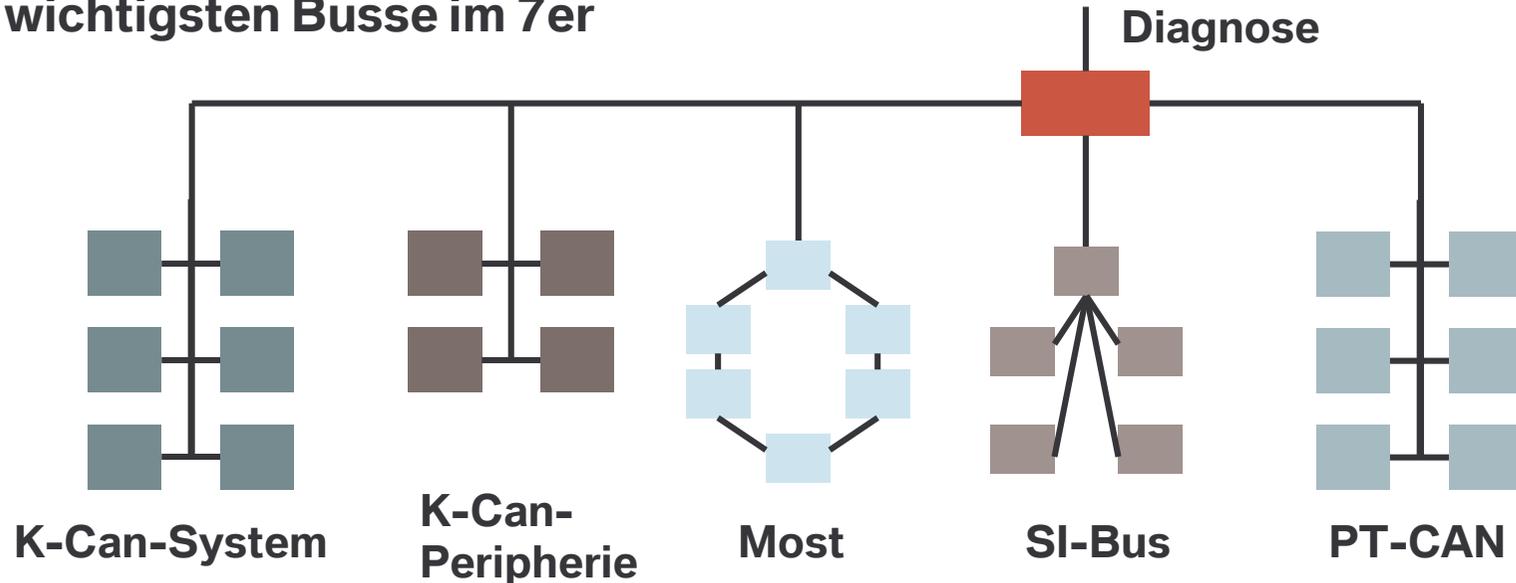
Prozesskette für Buskommunikation

Übersicht über wichtige Bustypen im Kfz

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge

Die wichtigsten Busse im 7er



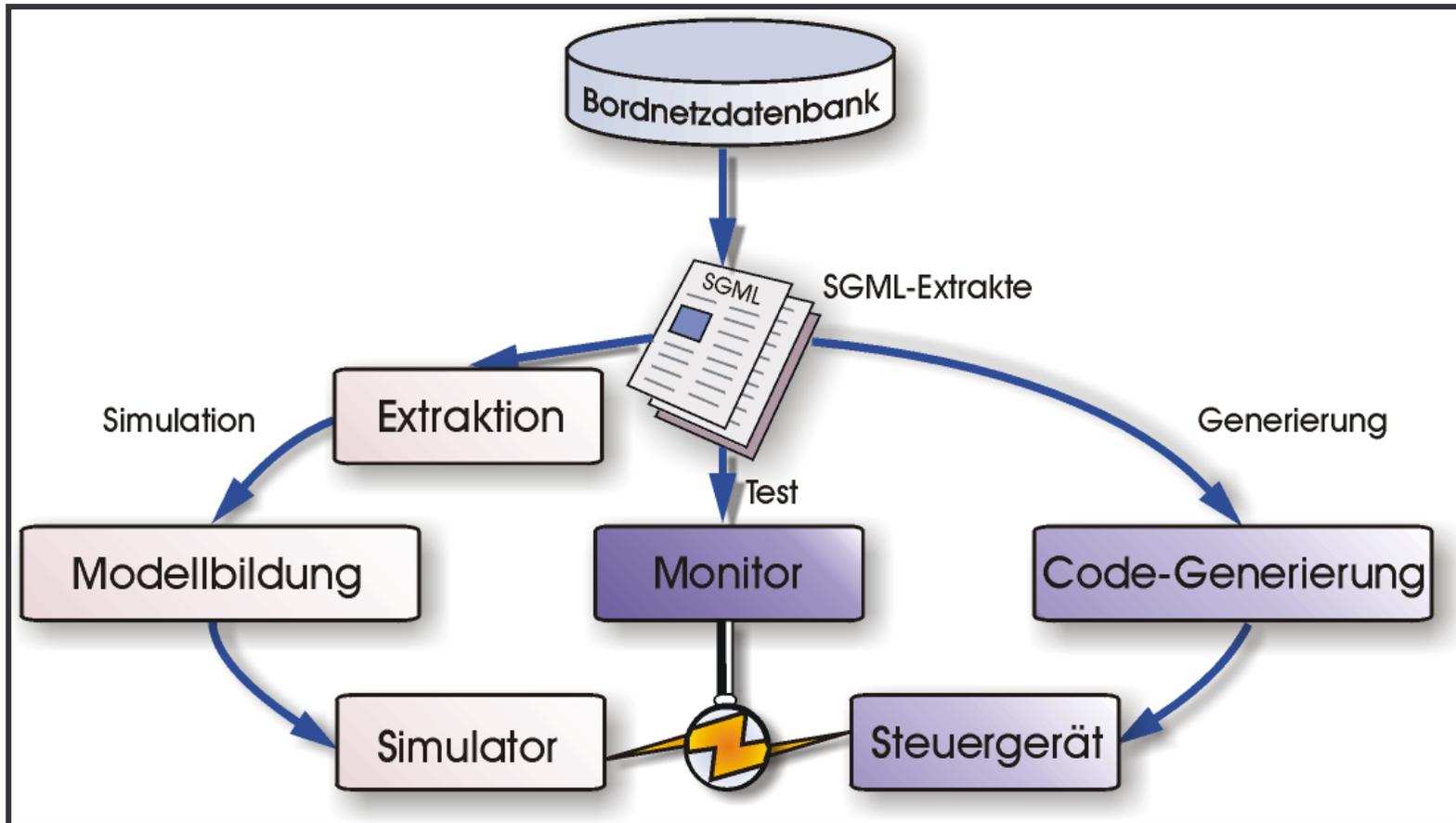
Bus	Name	Übertragungsrate	phys. Medium	Topologie	Anwendungsbereich
CAN	Controler Area Network	100/500 KBaud, asynchron	Cu-2 Draht	Netz (Multimaster)	Karosserie, Peripherie, Motor
Most	Media Oriented Systems Transport	bis 20 MBaud synchron/asynchron mit seperatem Kontroll- und steuerkanal	Lichtleiter	Ring (Master)	Multimedia-, Kommunikations- und Telematikanwendungen
SI	Sicherheitsbus (Byteflight)	10 MBaud, synchron/asynchron	Lichtleiter	Stern (mit Sternkoppler)	Sicherheitsanwendungen (Airbag)
Diagnose	KWP2000	115KBaud (9.6KBaud)	Cu-Seriell	P2P	Diagnose

Prozesskette für Buskommunikation

Übersicht

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge

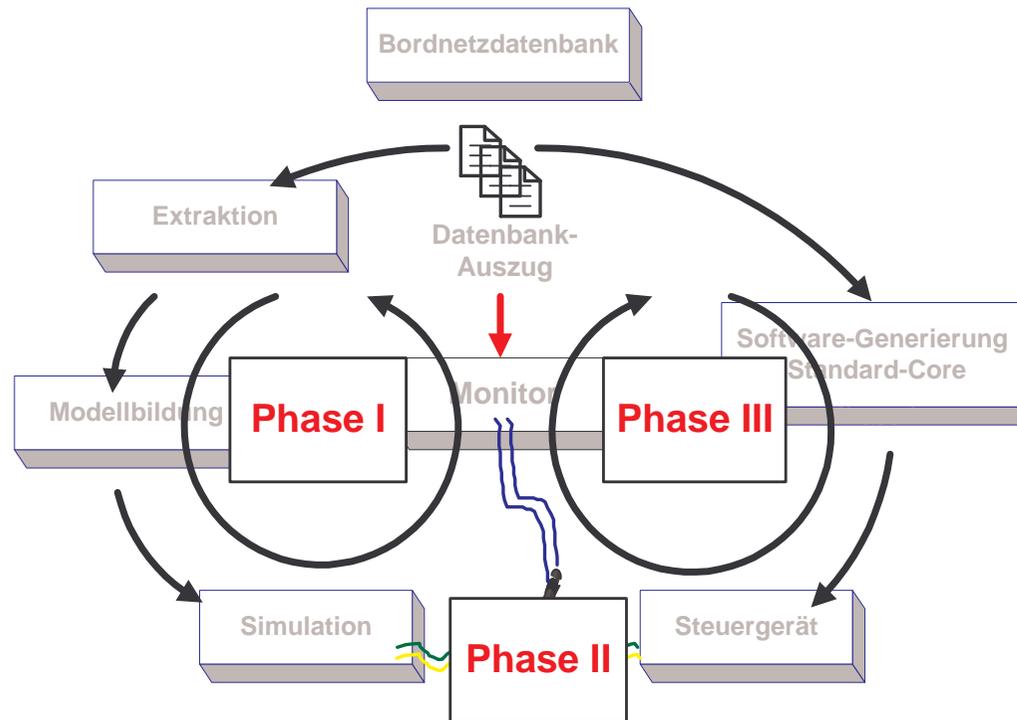


Konsistenzprüfung der Datenbankauszüge, Steuergeräte-kommunikation und Prüfung der Kommunikation im Gesamtsystem

Prozesskette für Buskommunikation – Prüfungen der Steuergerätekommunikation

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge



- Phase I: Konsistenzprüfung der Datenbankauszüge
Phase II: Prüfung der Steuergeräte-Kommunikation
Phase III: Prüfung der Kommunikation im Gesamtsystem

Restbussimulation für CAN-Steuergeräte – Merkmale des CAN-Protokolls

Gliederung:

1. Entwicklungskette
2. Prozesskette
Buskommunikation
3. Restbussimulation
4. Busmonitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere
Werkzeuge

Wesentliche Merkmale des CAN (Controller-Area-Network):

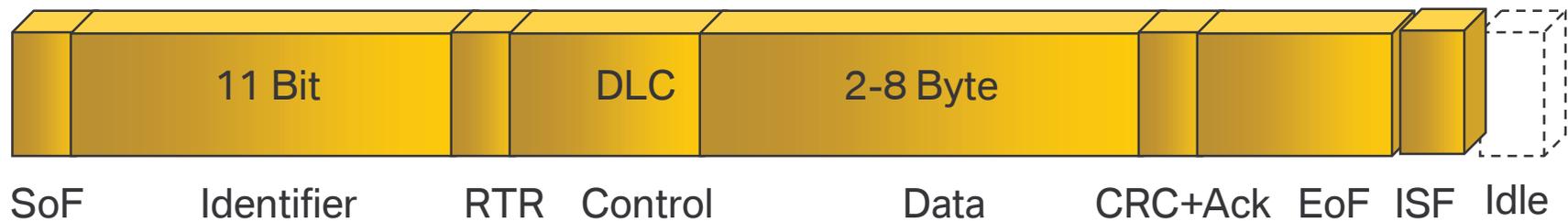
- ⇒ Genormtes Protokoll (ISO 11898 und ISO 11519-2)
- ⇒ linienförmige Bustopologie (Länge bis zu ca. 1000m je nach Datenrate)
- ⇒ Multi-Master System
- ⇒ Verfahren der bitweise verlustfreien Arbitrierung -> "Niedrigere ID gewinnt"
- ⇒ bis zu 1 Mbit/s Datenrate (netto bis zu ca. 500 kBit/s)
- ⇒ weitreichende Mechanismen zur Fehlererkennung (Hammingdistanz 6 (2))
- ⇒ $< 4.5 * 10e-13$ Wahrscheinlichkeit für die Nichterkennung übertragener Fehler

Restbussimulation für CAN-Steuergeräte – Merkmale des CAN-Protokolls

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge

Grundlegender Aufbau einer Can-Botschaft (nicht extended):



Gesamte Länge (ca.): 60..130 Bit

Einzelheiten zu CAN bei:

Konrad Etchberger, Controller-Area-Network, Hanser-Verlag, ISBN 3-446-19413-2

Wolfhard Lawrenz, Controller Area Network, Hüthig Verlag, ISBN 3-7785-2780-0

Restbussimulation für CAN-Steuergeräte – Definition

Definition:

Gliederung:

1. Entwicklungskette
2. Prozesskette
Buskommunikation
3. Restbussimulation
4. Busmonitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere
Werkzeuge

“Restbussimulation” oder “Simulation der Kommunikationsumgebung” ist die Darstellung des Kommunikationsverhaltens von Steuergeräten im Fahrzeug.

- ⇒ Es wird hauptsächlich das Kommunikationsverhalten abgebildet; Funktionales Verhalten kann automatisiert nur teilweise abgebildet werden.
- ⇒ Wichtig ist daher, ein offenes und für den Entwickler erweiterbares System zu schaffen.
- ⇒ Gemischter virtuell/realer Betrieb muss möglich sein;

Restbussimulation für CAN-Steuergeräte – Ansatz bei BMW

Gliederung:

1. Entwicklungskette
2. Prozesskette
Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere
Werkzeuge

⇒ Automatische Generierung der benötigten Programme und Datenfiles aus den Informationen in der BN-Datenbank

Vorteile:

⇒ Es ist immer der aktuelle Stand des Bordnetzes darstellbar.

⇒ Fehler, die bei manueller Programmierung auftreten, werden ausgeschlossen

⇒ Fehlerhafte Einträge und fehlende Daten in der BN-Datenbank können frühzeitig erkannt werden.

Restbussimulation für CAN-Steuergeräte – Ausprägungen

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbussimulation
4. Busmonitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge

⇒ **Restbussimulation für den Entwickler/Lieferanten:**

Umgebung (Steuergeräte) die direkt mit dem Steuergerät des Entwicklers kommunizieren, wird abgebildet

à maßgeschneiderte Lösung (1 reales, n virtuelle Komp.)

⇒ **Restbussimulation für die Systemintegration:**

à Einzelne Can-Busse werden partiell oder gesamthaft abgebildet, virtuelle Steuergeräte werden, wenn vorhanden durch reale ersetzt.

⇒ Für gesamthafte Simulationen mehrerer Busse (K-Can System, K-Can Peripherie, PT-Can) sind mehrere Simulatoren (Rechner) notwendig, die Gateways müssen entweder real, oder als separates Simulationsmodell vorliegen.

Restbussimulation für CAN-Steuergeräte – Umsetzung

Telegrammklassen:

Mechanismen für die Versendung von Botschaften (Telegrammen, Nachrichten) auf dem Can-Bus

Vorteile:

⇒ Einhaltung von “Spielregeln” macht den Datenverkehr auf dem Bus berechenbarer (z.B. Latenzzeiten, Buslasten)

3 Haupt-Klassen:

⇒ zyklisch

⇒ ereignisgesteuert

⇒ Anfrage

⇒ Weitere Telegrammklassen bei Bedarf (z.B. zyklisch mit Stoppbotschaft)

Gliederung:

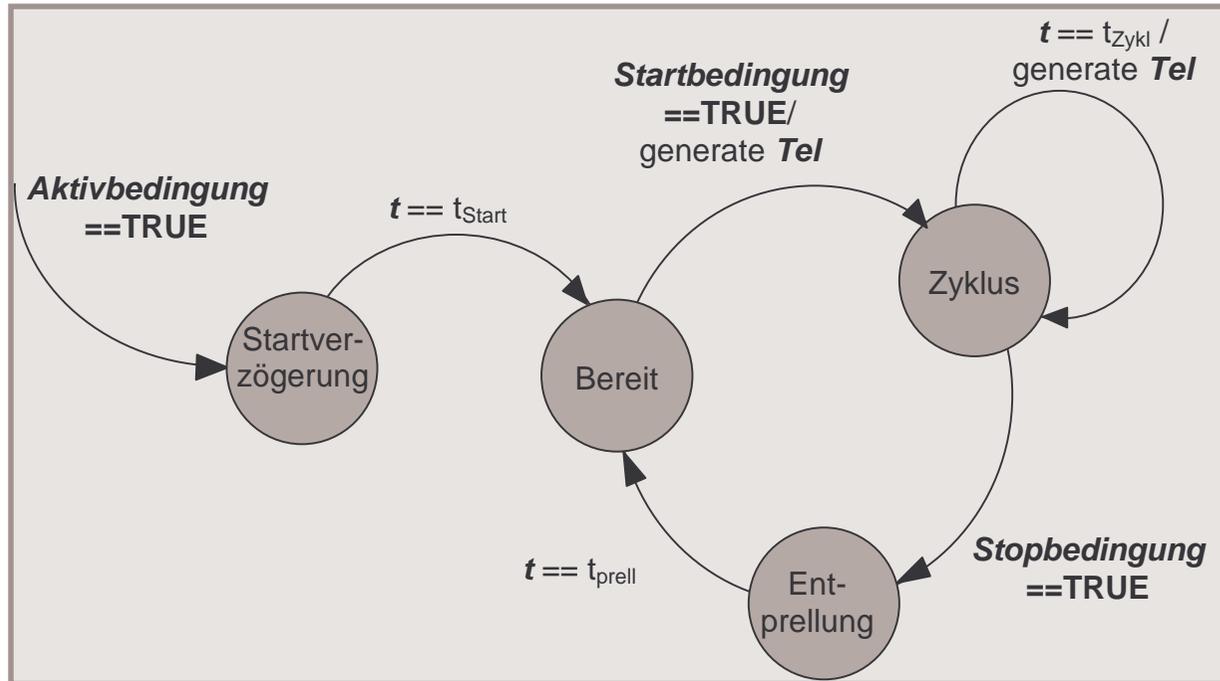
1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbussimulation
4. Busmonitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge

Restbussimulation für CAN-Steuergeräte – Umsetzung

Beispiel: Telegrammklasse für die Versendung zyklischer Botschaften

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge



Parameter: Activcondition*, Start-Delay-Time*, Startcondition, Tcycle*, Stopcondition, Debounce-Time* (Entprellzeit)

(* Repräsentation in der Datenbank)

Restbussimulation für CAN-Steuergeräte

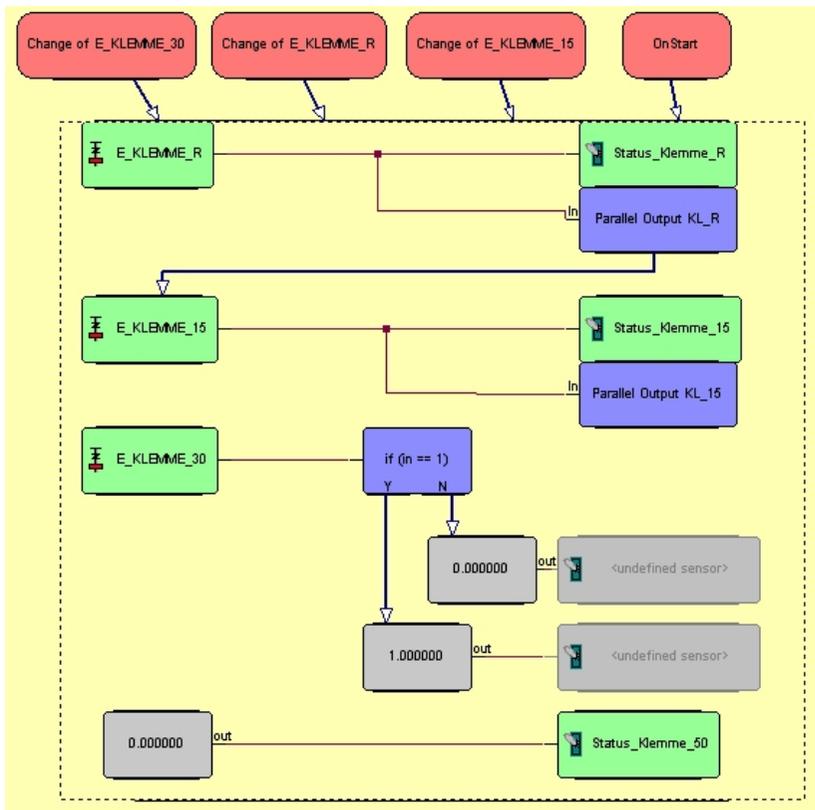
System- und Kundenfunktionen

Beispiel für Kundenfunktionen: Klemmenstatus

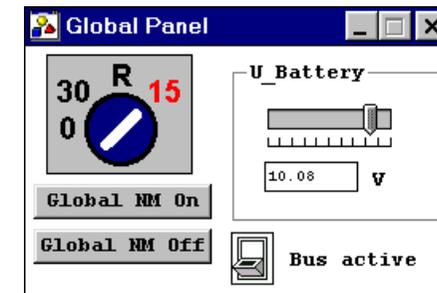
Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge

Drehung am Zündschlüssel oder Betätigung des Start-Stop Knopfes führt zu Signaländerungen in der korrespondierenden CAN-Botschaft



Klemmenstatus-Automat
(stark vereinfacht)



Simulationspanel

Restbussimulation für CAN-Steuergeräte

System- und Kundenfunktionen

Beispiel für Systemfunktionen: Netzwerkmanagement

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge

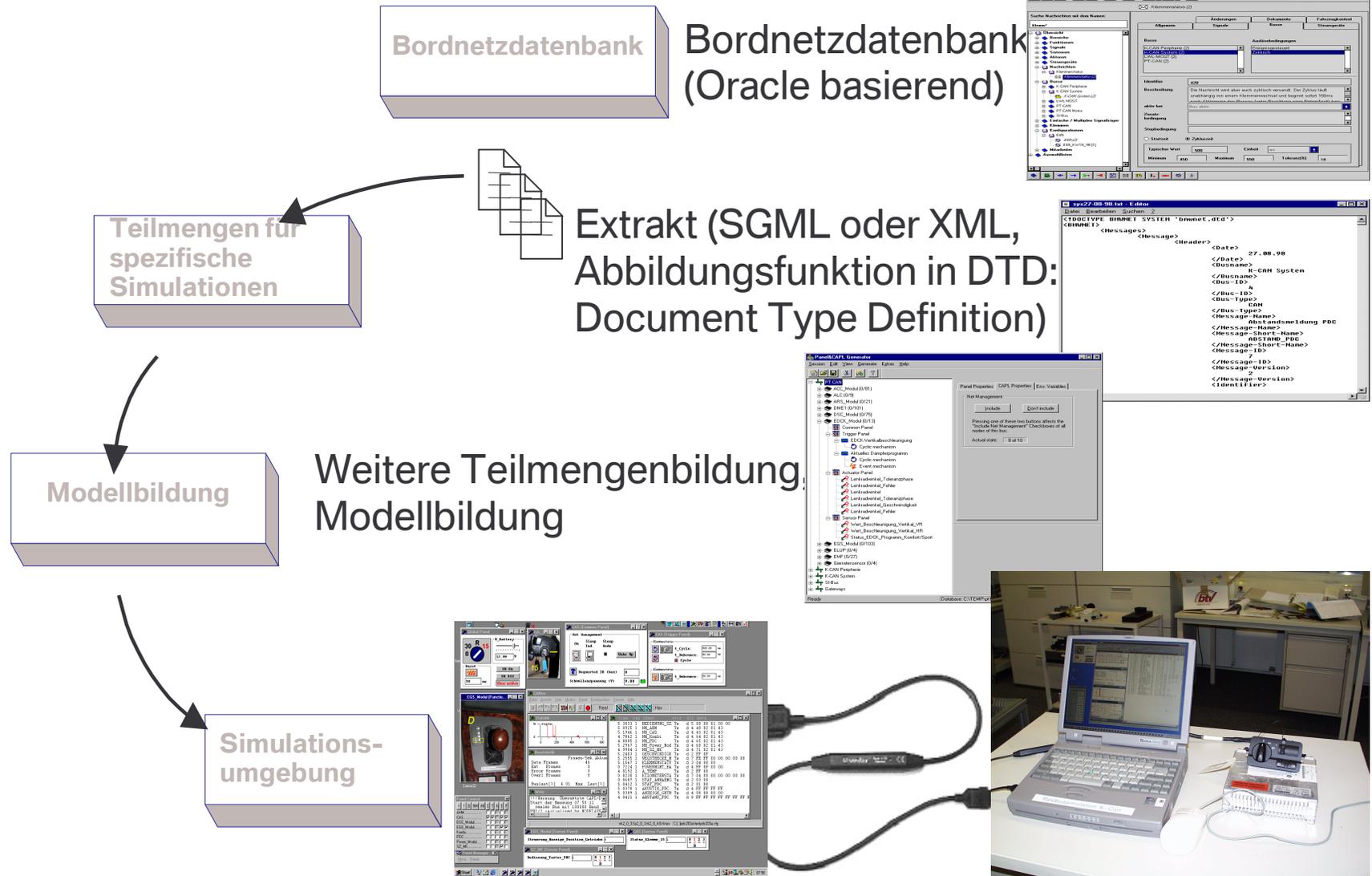
- Zwischen den Steuergeräten werden spezielle Statusbotschaften weitergereicht (Token), diese signalisieren allen Komponenten generelle Stati über Zustände einzelner Teilnehmer (z.B. Schlafbereitschaft, Sleep Indication); Es bildet sich ein logischer Ring aus.
- Es gibt Regeln für das Aufstarten, Einklinken in den logischen Ring, Einschlafen usw.

Einzelheiten zum Netzwerkmanagement bei <http://www.osek-vdx.org>

Restbussimulation für CAN-Steuergeräte – Umsetzung

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge



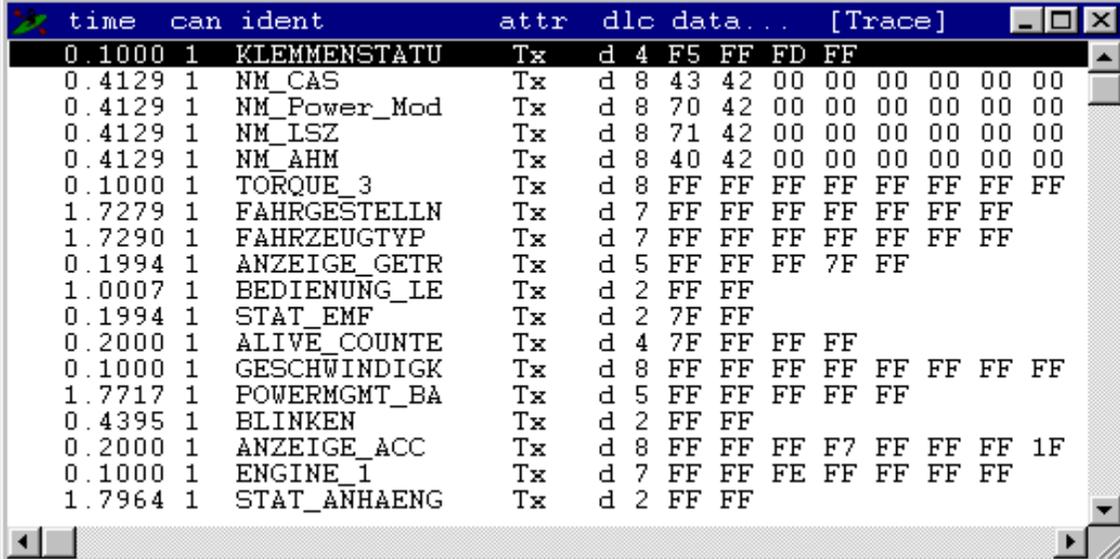
Busmonitoring – am Beispiel CAN

Grenzen bisheriger Aufzeichnungswerkzeuge

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge

- ⇒ Listendarstellung à Fehler (speziell temporäre) in Signalinhalten sind kaum auffindbar
- ⇒ Keine Prüfung der Sendebedingungen und weiterer für das Gesamtsystemverhalten wesentlicher Parameter
- ⇒ Keine Visualisierung des Gesamtsystemverhaltens, z.B. Netzwerkmanagement



time	can id	ident	attr	dlc	data
0.1000	1	KLEMMENSTATU	Tx	d 4	F5 FF FD FF
0.4129	1	NM_CAS	Tx	d 8	43 42 00 00 00 00 00 00
0.4129	1	NM_Power_Mod	Tx	d 8	70 42 00 00 00 00 00 00
0.4129	1	NM_LSZ	Tx	d 8	71 42 00 00 00 00 00 00
0.4129	1	NM_AHM	Tx	d 8	40 42 00 00 00 00 00 00
0.1000	1	TORQUE_3	Tx	d 8	FF FF FF FF FF FF FF FF
1.7279	1	FAHRGESTELLN	Tx	d 7	FF FF FF FF FF FF FF
1.7290	1	FAHRZEUGTYP	Tx	d 7	FF FF FF FF FF FF FF
0.1994	1	ANZEIGE_GETR	Tx	d 5	FF FF FF 7F FF
1.0007	1	BEDIENUNG_LE	Tx	d 2	FF FF
0.1994	1	STAT_EMF	Tx	d 2	7F FF
0.2000	1	ALIVE_COUNT	Tx	d 4	7F FF FF FF
0.1000	1	GESCHWINDIGK	Tx	d 8	FF FF FF FF FF FF FF FF
1.7717	1	POWERMGMT_BA	Tx	d 5	FF FF FF FF FF
0.4395	1	BLINKEN	Tx	d 2	FF FF
0.2000	1	ANZEIGE_ACC	Tx	d 8	FF FF FF F7 FF FF FF 1F
0.1000	1	ENGINE_1	Tx	d 7	FF FF FE FF FF FF FF
1.7964	1	STAT_ANHAENG	Tx	d 2	FF FF

Busmonitoring – am Beispiel CAN

CanBusMonitor

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge

Ziele des CanBusMonitors:

- Visualisierung
- Parametrierung aus der Bordnetzdatenbank
- Darstellung von Systemeigenschaften (Netzwerkmanagement)
- Ja / Nein – Entscheidung
- Logging
- Modularität und Erweiterbarkeit

Anforderungen:

- Laptop oder PC mit PCMCIA-Schacht
- Hardware Can-Zugang mit Treiber und API

Busmonitoring – am Beispiel CAN

CanBusMonitor – Oberfläche

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge

The screenshot displays the CanBusMonitor (CBM) software interface. The main window, titled 'K-Can_4_1_0.cbm - CBM', shows a network diagram with various nodes and connections. A legend on the left indicates node states: Alive (purple), Ring (green), Sleep (blue), and Default (white). It also shows symbols for Sleep Indication, Sleep Ack., Node inactive, Node active, and Error in node. The diagram includes nodes for 'K-CAN Peripherie' (SM_F, SM_B, TM_BI, TM_F, Power, TM_BI) and 'K-CAN System' (CAS, BZM, FKA, IHKA, PDC, FB_IH, LSZ, CIM, Lufttec, Regen, Wisch, DWA, AHM, SH_ZH, SHD). A red dotted line highlights the 'LWL-MOST' connection between the 'K-CAN Peripherie' and 'K-CAN System' nodes.

An inset window titled 'CAS' provides details for a selected message. It shows the message amount (228), percent (18%), cyclic count (225), event count (0), request (0), and ID (0H). The message list includes 'Klemmenstatus' (ID 130H) and other messages. The active bus activity is shown as 'Bus act' with a minimum of 90ms and a maximum of 110ms. The parameter table below shows the status of various components:

Parameter	Value
Status_Schluessel_S...	Schluessel steckt im ZAS
Status_Schluessel_T...	Signal ungueltig
Status_Schluessel_...	Signal ungueltig
Status_Keylock	Signal ungueltig
Status_Klemme_15	Klemme 15 EIN
Status_Klemme_50	Signal ungueltig

Details

Busmonitoring – am Beispiel CAN

CanBusMonitor – Weitere Eigenschaften

Gliederung:

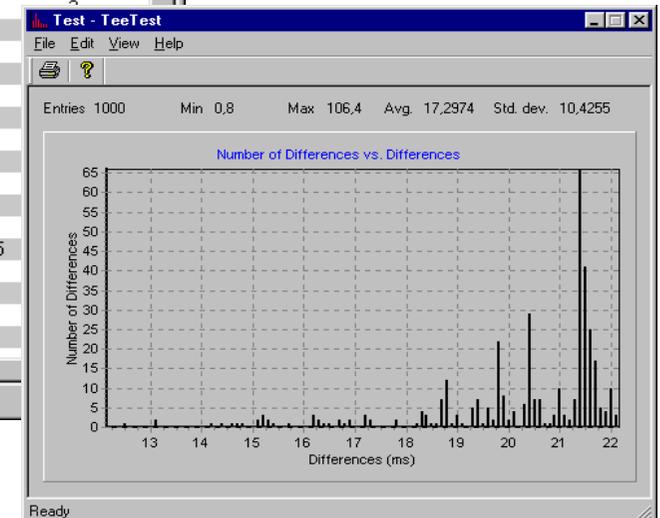
1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge

Date	Time (ms)	ID	Name	Clomp	DLC	Data	Info
Di 05. Okt 1999	92,5	487	BEDIENUNG_SITZH...	15	2	0F 00 00 00 00 00 00 00	
	92,6	488	BEDIENUNG_SITZH...	15	2	0F 00 00 00 00 00 00 00	
	92,6	491	BEDIENUNG_AKTIV...	15	2	03 00 00 00 00 00 00 00	
	92,5	492	BEDIENUNG_AKTIV...	15	2	03 00 00 00 00 00 00 00	
	95,1	764	STAT_ZV_KLAPPEN	15	7	FF FF FF FF 7F 00 00 00	
	57157,1	1248	NM_Kombi	15	8	63 42 00 00 00 00 00 00	
	100,6	468	STAT_KOMBI	15	7	FF 3F 1F 0F FF 3F 7F 00	
	194,3	530	HOEHENSTAND_LU...	15	6	FF F7 7F 00 00 00 00 00	S
	101,8	534	HOEHENSTAND_L...	15	6	00 F0 7F 00 F0 7F 00 00	S
	101,8	546	NEIGUNG_FC	15	2	FF 03 00 00 00 00 00 00	
	101,8	211	FERNBEDIENUNG...	15	4	FF FF 3F 03 00 00 00 00	
	101,8	500	FERNBEDIENUNG...	15	2	07 00 00 00 00 00 00 00	
	101,8	459	BEDIENUNG_EC	15	2	07 00 00 00 00 00 00 00	

Signalfenster

Name	Value	Raw	Min	Max
Spannung_Batterie	0			
Status_Standverbraucher	0			
Steuerung_Unterspannung...	0			
Anforderung_Normalspannung	Signal ungültig	3	3	
Geschwindigkeit_Fahrzeug	0			
Status_Crashschwere	0			
Status_Klemme_15	Klemme 15 EIN	1	1	
Status_Klemme_R	Klemme R EIN	1	1	
Status_Motor_Läuft	0			
Status_Parklicht_Links	AUS	0		
Status_Parklicht_Rechts	AUS	0		
Status_Schlüssel_Steckt	Signal ungültig	3	3	
Status_Schlüssel_Typ	Signal invalid	15	15	
Status_Standlicht	AUS	0		
Status_Wegfahrsperr	Signal ungültig	7	7	

Histogramme



Logging

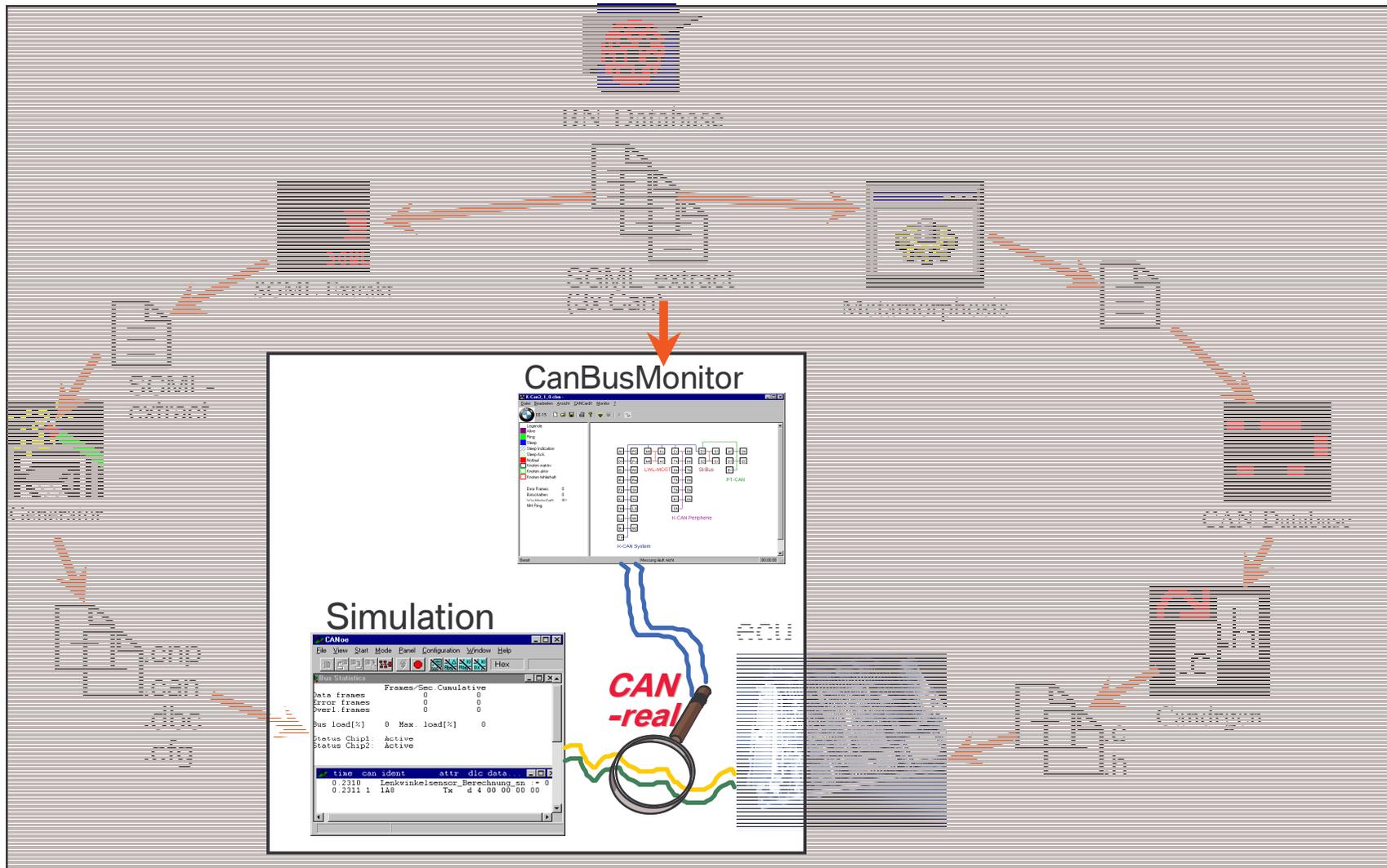
- Fehlerdokumentation
- Replay

„Problem“: „Endliche Auflösung“ bei der Fehlerdetektion

Restbussimulation – Busmonitoring Demo

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge



Testplattformen für das Gesamtsystem

**Für den Gesamtsystemtest des Systems Elektrik/Elektronik
kann man sich prinzipiell 4 Plattformen vorstellen:**

Gliederung:

1. Entwicklungskette
2. Prozesskette
Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere
Werkzeuge

1. Rein virtuelle Plattform
2. Plattform mit realen Komponenten (unabhängig von geometrischen Anordnung)
3. Plattform mit realen Komponenten (lageabhängig, nicht fahrbar)
4. Prototypfahrzeuge

Testplattformen für das Gesamtsystem

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge

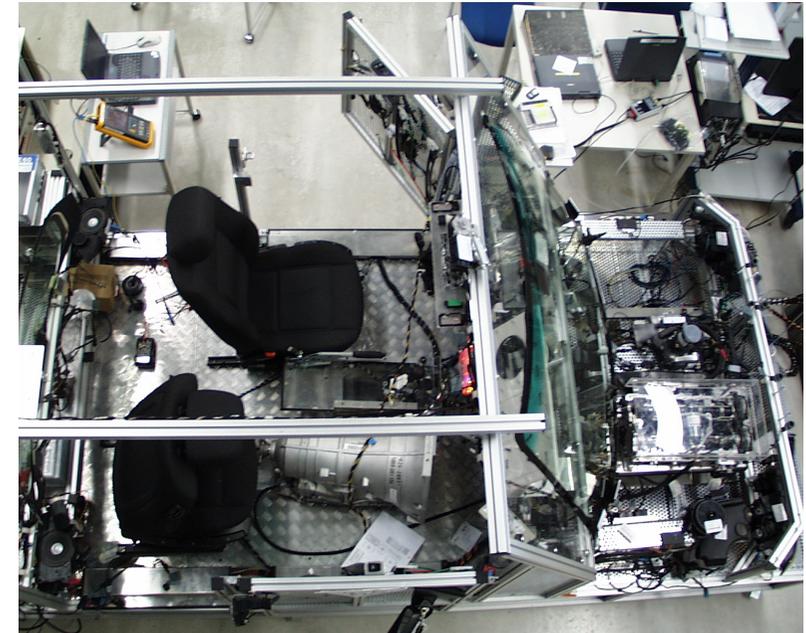
- zu 1.: Eine rein virtuelle Plattform für ein Elektronik-Gesamtsystem ist heute technisch nicht realisierbar (u.A. Echtzeitanforderungen, abstrahierbare, vollständige, granulare und automatisierbare Funktionsbeschreibung). Sie würde außerdem einen stark veränderten Entwicklungsprozess erfordern (zeitliche Entkoppelung)
- zu 2.: Eine Plattform ohne geometrischen Bezug wurde und wird von verschiedenen Seiten immer wieder versucht, allerdings ergeben sich aus einem „generischen Kabelbaum“ zu viele nicht beherrschbare Fehlerquellen
- zu 3.: mit dem Laborfahrzeug wird bei BMW dieser Weg seit vielen Jahren beschritten
- zu 4.: solche Fahrzeuge sind bei allen Herstellern im Einsatz

Testplattformen für das Gesamtsystem

Laborfahrzeug statisch

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge



Aufgaben (u.A.):

- Erste Integration im Gesamtsystem
- Test auf Systemverträglichkeit (Systemfunktionen)
- Erste Abbildung von Werksprozessen
- Erster Kabelbaumcheck

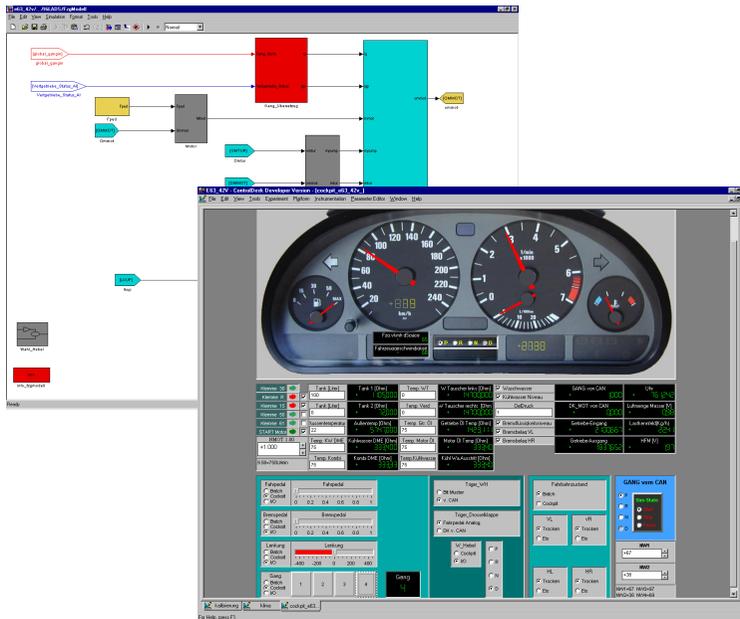
Testplattformen für das Gesamtsystem Laborfahrzeug – quasidynamischer Betrieb mit HIL

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge

HIL-Umgebung (DSpace mit Mehrspurmodell in Matlab/Simulink)

- Eingangsschnittstellen vom Laborfahrzeug u.A.: physikalische Signale (z.B. Drosselklappenwinkel, Gaspedalwinkel usw.) und Bussignale (z.B. CAN-Signal „eingelegter Gang“)
- Ausgangsschnittstellen u.A. Radsignale, Temperatursignale, weitere
- Damit sind (in eingeschränktem Maße) erste dynamische Test möglich



Testplattformen für das Gesamtsystem Laborfahrzeug dynamisch

Gliederung:

1. Entwicklungskette
2. Prozesskette
Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere
Werkzeuge



Aufgaben (u.A.):

- Erster gesamthafter Test von Systemfunktionen unter dynamischen Bedingungen;
Test auf System-verträglichkeit in Verbindung mit Temperaturen
- Überprüfung anhand von Checklisten am Fahrzeug auf Inhalt und Praktikabilität
- Robustheitstests
- Zurzeit einzige Alternative zu HiL

Testplattformen für das Gesamtsystem

FIT - Fahrzeugintensivtest

Gliederung:

1. Entwicklungskette
2. Prozesskette
Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere
Werkzeuge



Aufgaben (u.A.):

- Tests unter realen Bedingungen
- Abarbeitung standardisierter Checklisten
- Breitentests unter Beteiligung mehrerer Fahrzeuge (Statistik, Streuung)
- Abdeckung von Varianten
- Fehlermonitoring und unmittelbare Kommunikation zum Entwickler (täglich)

Gesamtsystemtests und -methoden

Grenzen der vollständigen Testbarkeit

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge

Sind vollständige Tests aller (erlaubten und verbotenen) Buskommunikationszustände möglich!

Antwort: Theoretisch ja, in der Praxis nein

Beispiel 7er BMW: ca. 1000 Bussignale mit je im Mittel 4Bit à etwa 10^{48} Kombinationsmöglichkeiten

Grenzen der Verarbeitungsgeschwindigkeit für Informationen:

$N \leq \frac{\pi E}{\hbar \ln 2}$ Folgt aus Bekensteins Grenze und der Lichtgeschwindigkeit

Mit $E = mc^2$ Ergibt sich für die maximale Rechengeschwindigkeit eines auf Materie basierenden Systems je Gramm ca. 10^{48} Bits/s

Hierbei sind aber geometrische Strukturen (z.B. Planck – Länge $\approx 1.6 \cdot 10^{-35} \text{ m}$) angenommen die um den Faktor 10^{20} kleiner sind als reale Strukturen (z.B. Größe des Protons)

à **Vollständige Prüfungen einer komplexen Buskommunikation auf Basis einer „Brute Force“ Methode sind und werden nicht möglich sein**

Gesamtsystemtests und –methoden

Auswege

Gliederung:

1. Entwicklungskette
2. Prozesskette
Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere
Werkzeuge

Man muss sowohl Maßnahmen anwenden um die die Komplexität von vornherein besser beherrschen zu können, als auch geeignete adaptive Tests durchführen:

Design to Test (Vorbeugung)

- Definition einer geeigneten Prozesskette vom Anforderungsmanagement bis zum verifizierten Code
- Modellierung (z.B. UML) wo möglich und automatische Prüfung (TTCN)
- Simulation im Vorfeld
- Automatische Codegenerierung (z.B. Standard-Core, Autosar)
- Codechecks
- Reduzierung von zu großen Abhängigkeiten der einzelnen Signale (definierte Sender-Empfänger Beziehungen)
- Geeignete Partitionierung von Funktionen auf Steuergeräten
- Akzeptanzfilter (Hardware) in den Steuergeräten (schwaches Mittel, da die Filter meistens offen sein müssen)

Gesamtsystemtests und –methoden

Auswege

Gliederung: Eingrenzung von Tests auf Problembereiche (Erfahrung):

1. Entwicklungskette

2. Prozesskette
Buskommunikation

3. Restbus-simulation

4. Bus-monitoring

5. Testplattformen

6. Gesamtsystemtests

7. Weitere
Werkzeuge

- Steuergeräte und Funktionen mit (partiell) sehr hohem Datenaufkommen und systembedingt hohem Vernetzungsgrad (z.B. Gateways)
- Testhierarchie mit Abstufung nach Folgen eines potentiellen Ausfalls (z.B. sicherheitskritisch, Liegenbleiber, Komforteinschränkungen)
- außergewöhnliche Systemzustände (z.B. hohe Buslast, Robustheitstests)
- Komplexe Systemzustände (z.B. Einschlafen und Aufwachen des Bordnetzes)
- Komplexe Hardwareabhängigkeiten (z.B. Kabelbaum)
- Prozesse von allgemeiner Wichtigkeit (z.B. Fahrzeugdiagnose, -update, -codierung, Werksprozesse)
- Datenbankbasierte Funktionstests, Testplanung mit Aufwandsabschätzung
- Steuergeräte und Funktionen, die in der Vergangenheit immer wieder zu Problemen geführt haben

Gesamtsystemtests und –methoden

Beispiele für Tests: Kabelbaum / Pinning

Gliederung: Warum ist der Kabelbaum im Bordnetz ein kritisches Element?

1. Entwicklungskette

2. Prozesskette Buskommunikation

3. Restbus-simulation

4. Bus-monitoring

5. Testplattformen

6. Gesamtsystemtests

7. Weitere Werkzeuge

- Er ist komplex (Beispiel 7er: 3,3km Länge, 395 Stecker, Anschlüsse an 590 Sensoren / Aktoren, Gewicht 30-35kg)
- Fehler im Pinning führen fast immer zu kritischen Fehlern (bis hin zum Brand)
- Fehler sind teilweise schwer zu finden
- Der Kabelbaum ist zumindest am Anfang noch zum großen Teil Handarbeit
- Größere Fehler führen zu gravierendem (teuren) Änderungsbedarf
- Zukünftig sind erweiterte automatisch parametrisierte Prüfungen auf Basis von datenbankgestützten Systemen denkbar → Problem: Daten müssen aktuell gehalten und gepflegt werden (Prozess)

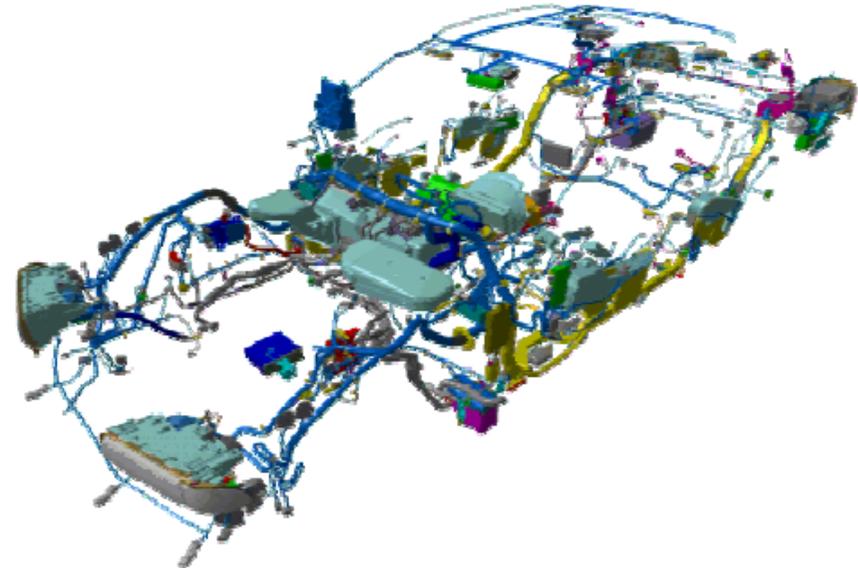
Gesamtsystemtests und -methoden

Beispiele für Tests: Kabelbaum / Pinning

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge

Bild Kabelbaumprüfung und Dokumentation am Laborfahrzeug



Elektrik / Elektronikkomponenten im 7er BMW

Gesamtsystemtests und -methoden

Beispiele für Tests: Einschlaf- und Wecktests

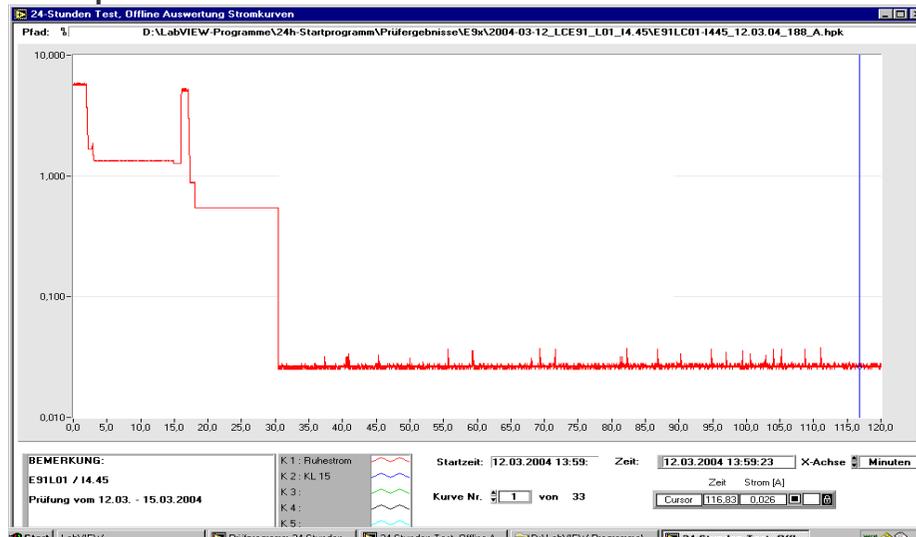
Gliederung: „Einschlafen“:

1. Entwicklungskette – Komplexer Systemzustand, korrekte Funktion ist wesentlich für den Ruhestrom und damit die Ladebilanz des Fahrzeugs
2. Prozesskette Buskommunikation – Alle Steuergeräte (Can-Busse) müssen ihr OK geben (Sleep Indication Flag)
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge

„Aufwachen“:

- Komplexer Systemzustand, Aufsynchronisation des „Netzwerkmanagement-Rings“ notwendig, Buslastspitzen, Gültigkeit von Signalinhalten für Partner wichtig

Beispiel für Tests: 24h Test



Gesamtsystemtests und -methoden

Beispiele für Tests: Einschlaf- und Wecktests

Gliederung: Beispiel für Tests: Dauerwecktest (Ansprechpartner Herr Mendelin)

1. Entwicklungskette
2. Prozesskette
Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere
Werkzeuge

Kurze Testbeschreibung:

- Der Dauerwecktest analysiert die CAN-Kommunikation und weckt zu stets variierenden Zeiten das Fahrzeug durch Senden von NM-Botschaften (wiederholtes Senden von Alive-Botschaften).
- Das Fzg. soll durch den Test für einen definierten Zeitraum wachgehalten werden.
- Ein kritischer Weckzeitpunkt liegt z.B. 1500ms nach Sleep-Acknowledge, da hier nach Ablauf des Timers $T_{\text{WaitBusSleep}}$ ein wichtiger Zustandsübergang in den meisten SG vorkommt.
- Der Test analysiert 2 CAN-Busse.
- Andere Bussysteme werden momentan durch Fehlerspeicherauswertungen (vorher-nachher-Vergleich) geprüft.

- Eine eingehende Analyse der Fehlerursachen und -auswirkungen ist nur zusammen mit den Entwicklern der Sender- und Empfänger-SG möglich.

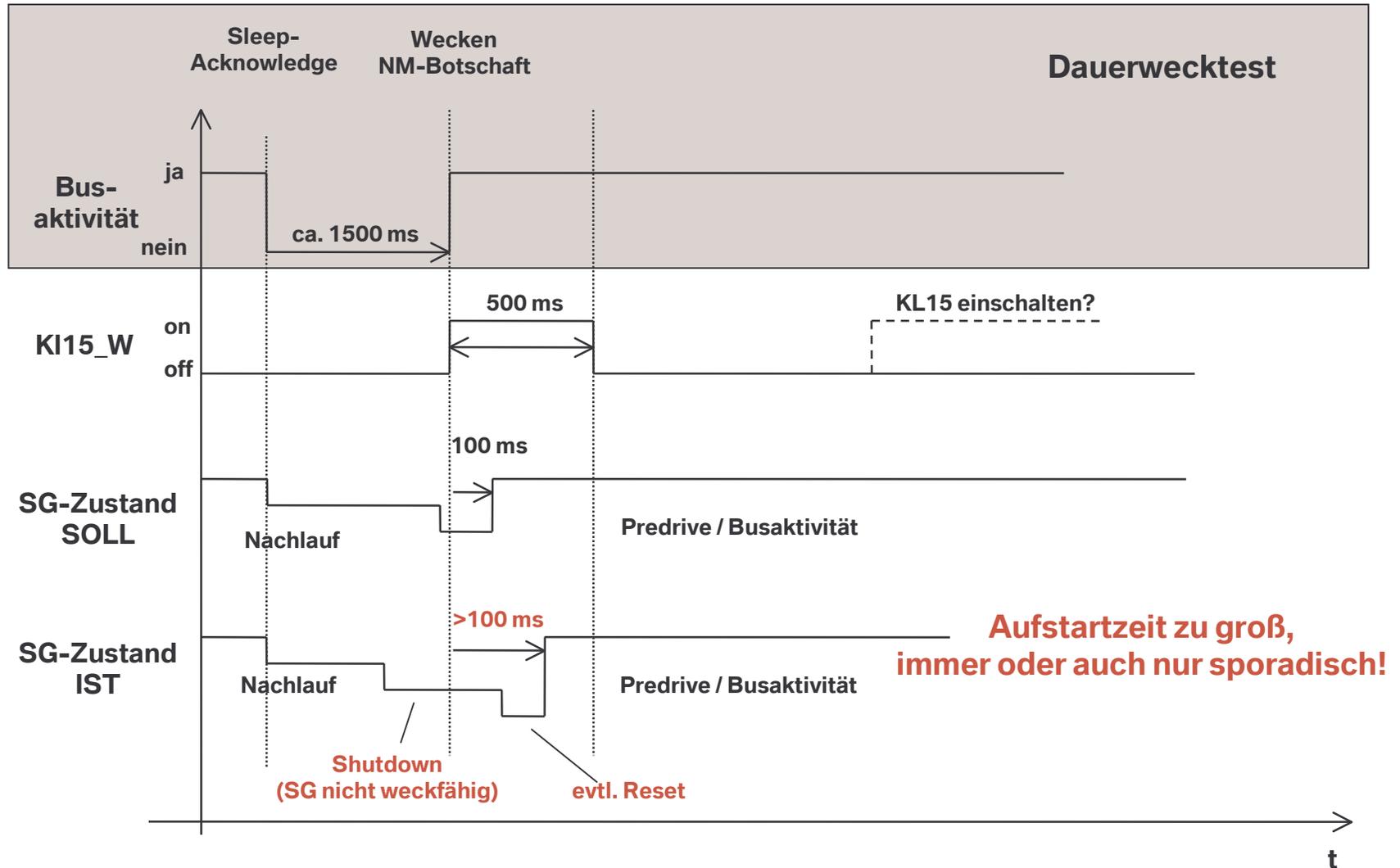
Gesamtsystemtests und -methoden

Beispiele für Tests: Einschlaf- und Wecktests

Beispiel für Tests: Dauerwecktest (Ansprechpartner Herr Mendelin)

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge



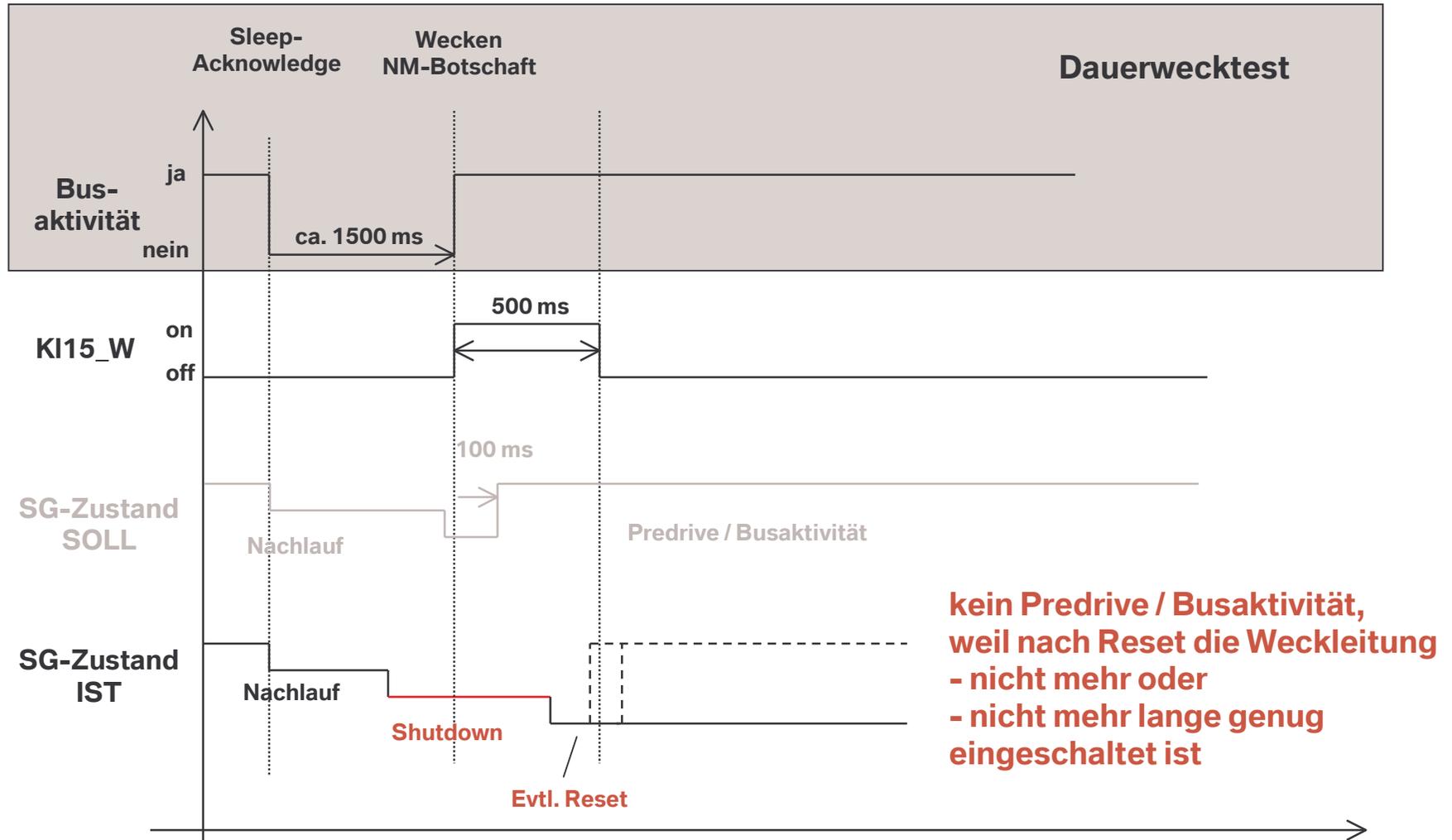
Gesamtsystemtests und -methoden

Beispiele für Tests: Einschlaf- und Wecktests

Beispiel für Tests: Dauerwecktest (Ansprechpartner Herr Mendelin)

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge



Gesamtsystemtests und -methoden

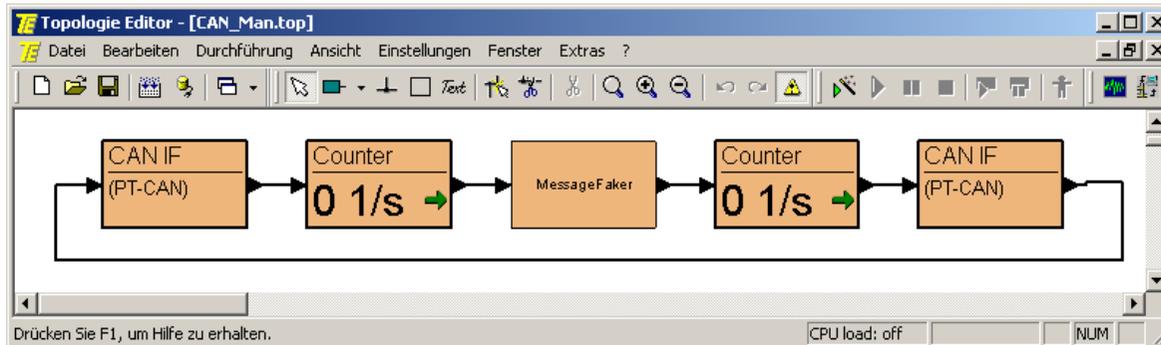
Beispiele für Tests: Robustheitstests / System FMEAs

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge

- Störungen (zeitlich begrenzt) im Bordnetz (physikalisch z.B. „Wackler“, informationstechnisch z.B. „Bitkipper“) sollen sich wenn möglich nur partiell auswirken und dürfen nicht zu sicherheitskritischen Zuständen führen
- Tests werden priorisiert nach Risiken, -> Risikofilter

Beispiel für Tests: Messagefaker



The screenshot shows the CANMan 1.4 software interface. The configuration panel includes:

- Eingabe neu: 73
- Bus: K-CAN System
- Botschaft: Steuerung Zentralverriegelung
- Signal: Steuerung_Zentralverriegelung_Tankklappe
- Fehlerart: Manuelle Eingabe
- Fehlerwert (Rohwert): 1
- Fehleranzahl: unbegrenzt

Buttons include "Daten übertragen" and "FehlerStart". The log window at the bottom contains the following text:

```
Enthält den aktuellen Anforderungsbefehl des ZV-Masters im C2
1111 = Signal ungültig
1-01 = Logisch entriegeln
1000 = Keine Änderung
0-11 = Sichern
0-10 = Verriegeln
0-01 = Entriegeln
0-00 = Elektrisch öffnen
Es wird das MessageFaker-Modul mit der UniqueID 73 verwenc
Es wird das MessageFaker-Modul mit der UniqueID 73 verwenc
Es wird das MessageFaker-Modul mit der UniqueID 73 verwenc
Es wird das MessageFaker-Modul mit der UniqueID 73 verwenc
```

Gesamtsystemtests und –methoden

Beispiele für Tests: Buslast-Tests

- Gliederung:
- Das Fahrzeug enthält diverse nichtdeterministische Bussysteme (z.B. CAN-Busse)
1. Entwicklungskette
- Aus Prinzip kann man hier nur in beschränktem Maße (abhängig von der Nachrichtenpriorität) Latenzzeiten garantieren
2. Prozesskette Buskommunikation
- Eine Verbesserung kann man durch eine sorgfältig geplante Buskommunikation erreichen (Design to Test)
3. Restbus-simulation
4. Bus-monitoring
- à Man sollte Grenzen für die Buslast (statisch, dynamisch) nicht überschreiten. Diese sind abhängig von der Toleranz der Busteilnehmer gegenüber Signalverzögerungen bzw. –ausfällen. Wesentlich hierbei sind die Burst-Belastungen (Totalblockade des Busses für eine gewisse Zeit)
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge

Beispiel für Tests: gezielte Stimulation und Fehlerverfolgung

Bild von Ergebnissen, ev. Video

Testwerkzeuge für das Elektronik-Gesamtsystem

Anforderungen

Gliederung:

1. Entwicklungskette
2. Prozesskette
Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere
Werkzeuge

Anforderungen an Systemmesstechnik (u.A.):

- Möglichst geringe Wechselwirkung mit dem Fahrzeug
- Genaue zeitliche Korrelation von verschiedenen Bussignalen notwendig
- Korrelation auch mit anderen physikalischen Größen notwendig (z.B. physikalische, akustische, optische Größen)
- Bei Bedarf „Wireless-Anbindungen“
- Flexibilität in den Konfigurationsmöglichkeiten
- Standardisierung für den breiten Einsatz
- Geeignete Austauschformate
- Differenzierung nach Einsatzgebieten (Preis, Größe, Stromverbrauch)

Testwerkzeuge für das Elektronik-Gesamtsystem

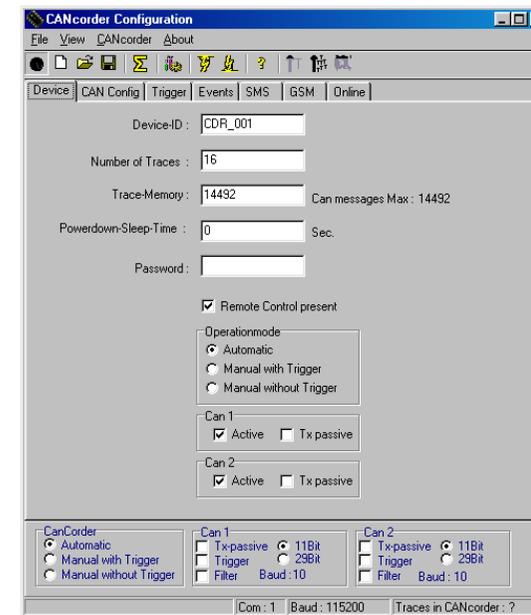
Beispiele: kleine Lösung

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge

„Kleine Messtechnik“

- Cancorder
- Bis zu 127 Trigger, derzeit bis zu 240 Tausend Botschaften
- 2 Can-Kanäle pro Gerät, meistens 2 Geräte parallel
- Einfache Triggerbedingungen, diverse weitere Konfigurationsmöglichkeiten
- SMS und GSM-Betrieb möglich
- Option zur korrelierten Aufzeichnung von anderen Bussen



Testwerkzeuge für das Elektronik-Gesamtsystem

Beispiele: große Lösung

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge

„Große Messtechnik“

- 1-2 fahrzeugtaugliche PC's
- Z.Zt. 4 Can-Kanäle, 1 Mostkanal mit Repeater
- Verschiedenste Triggermöglichkeiten
- Versorgung über Messbatterie und Ladegerät (Batteriezusaltung)
- WLAN
- Software CARMEN (Car Measurement Environment) für korrelierte Busaufzeichnungen und anwenderspezifische Auswertungen



Testwerkzeuge für das Elektronik-Gesamtsystem

Beispiele: große Lösung

Gliederung:

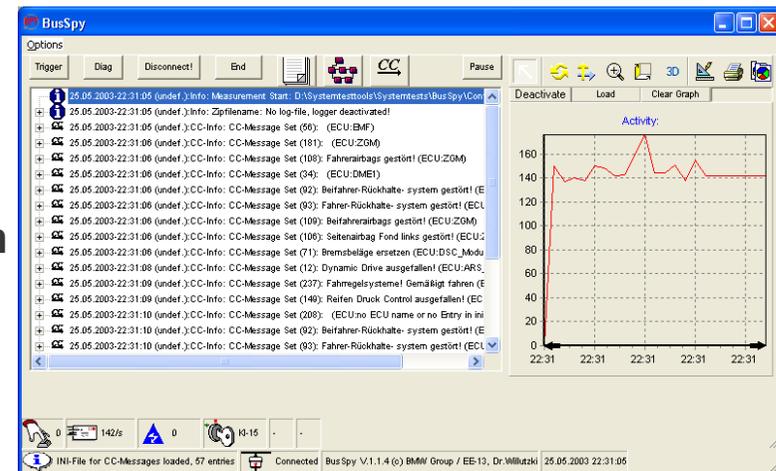
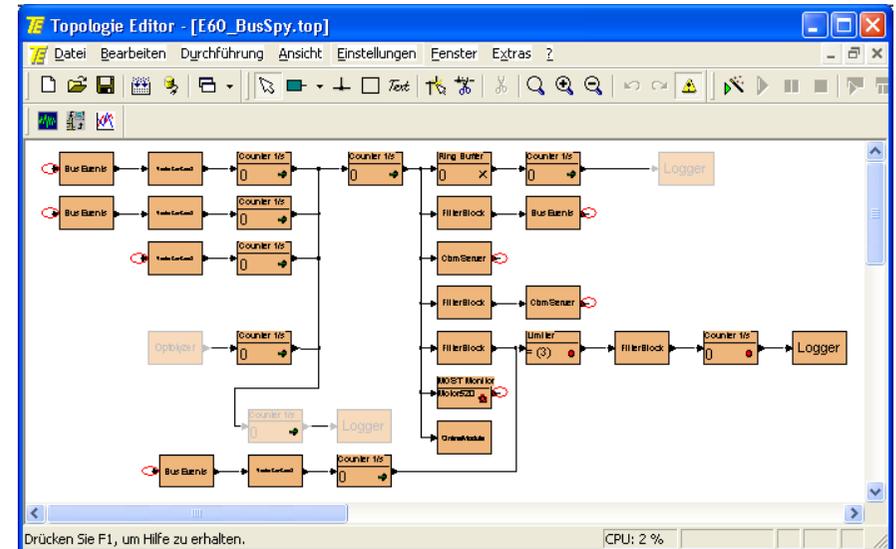
1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge

CARMEN bestehend aus:

- Framework Topologieeditor (Toped)
- Module (DLLs in process)
- Plugins (Datenbankzugriffe, Oberflächen (in/ex process))
- Automatisierungs-Clients (z.B. in Visual Basic, ex/in process)
- Plattform XP
- Komfortable Betrachter für korrelierte Messungen (mit Filter etc.)
- Most-Visualisierungen
- Signal Plotter
- Bordnetzdatenbank-Unterstützung

Diverse Systemtests (z.B. BusSpy) im Dauereinsatz an den dyn. LabCars

- „Fahrtenbuch“ für Busereignisse
- Trigger bei Bedarf
- Diagnoseabfragen für Systemtests



Testwerkzeuge für das Elektronik-Gesamtsystem

Beispiele: Videomesstechnik

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge

DART Videomesstechnik:

- Videoaufzeichnungssystem mit Ringspeicher
- Anbindung an bestehende große Messtechnik -> Korrelierte Aufzeichnung von Datenbussen und Video möglich



Testwerkzeuge für das Elektronik-Gesamtsystem

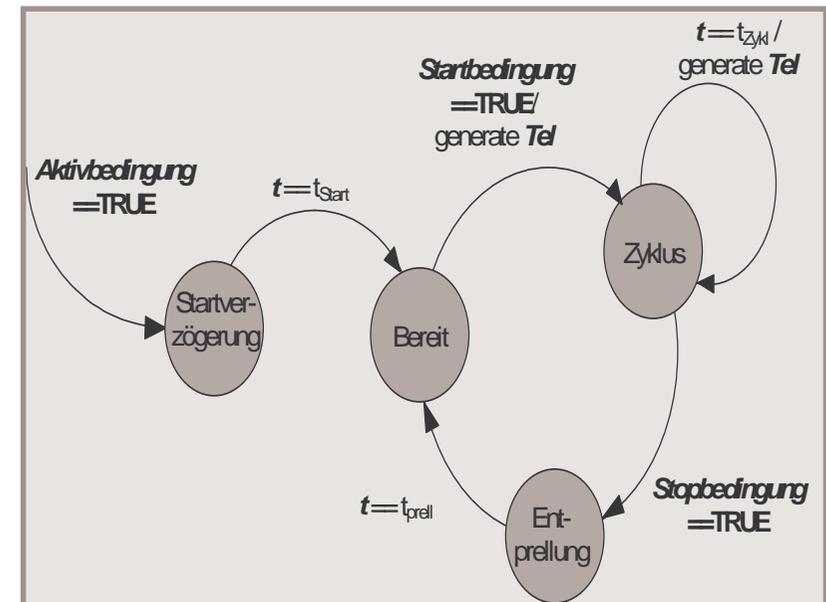
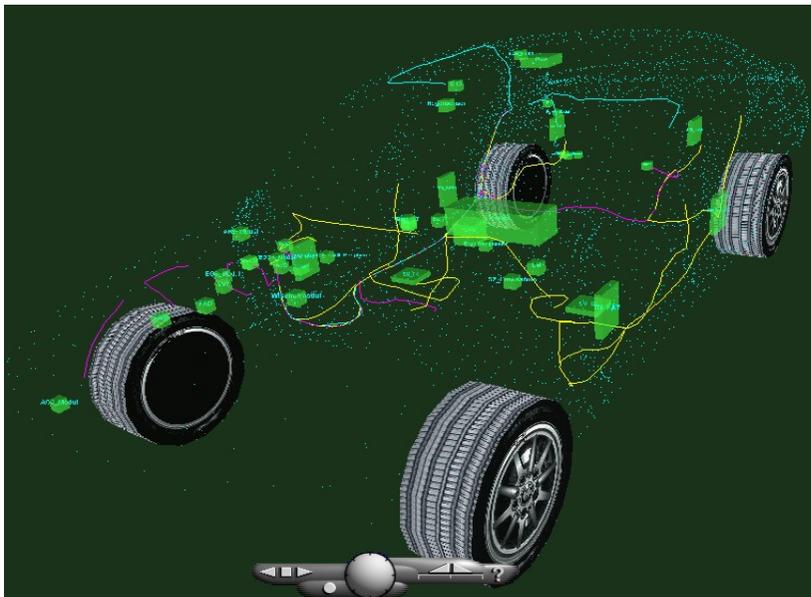
Visionen

Zukünftige mögliche Entwicklungen:

- Verteilung von Daten über Netzwerk à Verteilte Auswertungen
- Kombination mit zyklischen Systemtest (Systemstimulationen)
- Anbindung von Handhelds, Tablet-PCs
- Drahtlose Verbindungen (bei leistungsfähigen Netzwerken)
- Weitergehende Zustandsvisualisierungen
- Simulation von Testcases
- Generation von Tests aus Modelldatenbanken

Gliederung:

1. Entwicklungskette
2. Prozesskette Buskommunikation
3. Restbus-simulation
4. Bus-monitoring
5. Testplattformen
6. Gesamtsystemtests
7. Weitere Werkzeuge



Vielen Dank für Ihre Aufmerksamkeit !

