



Antriebs- und Fahrwerktechnik



Absicherung von Steuerungssoftware für Hybridsysteme

- Automatisierte Methode zur Testfallgenerierung

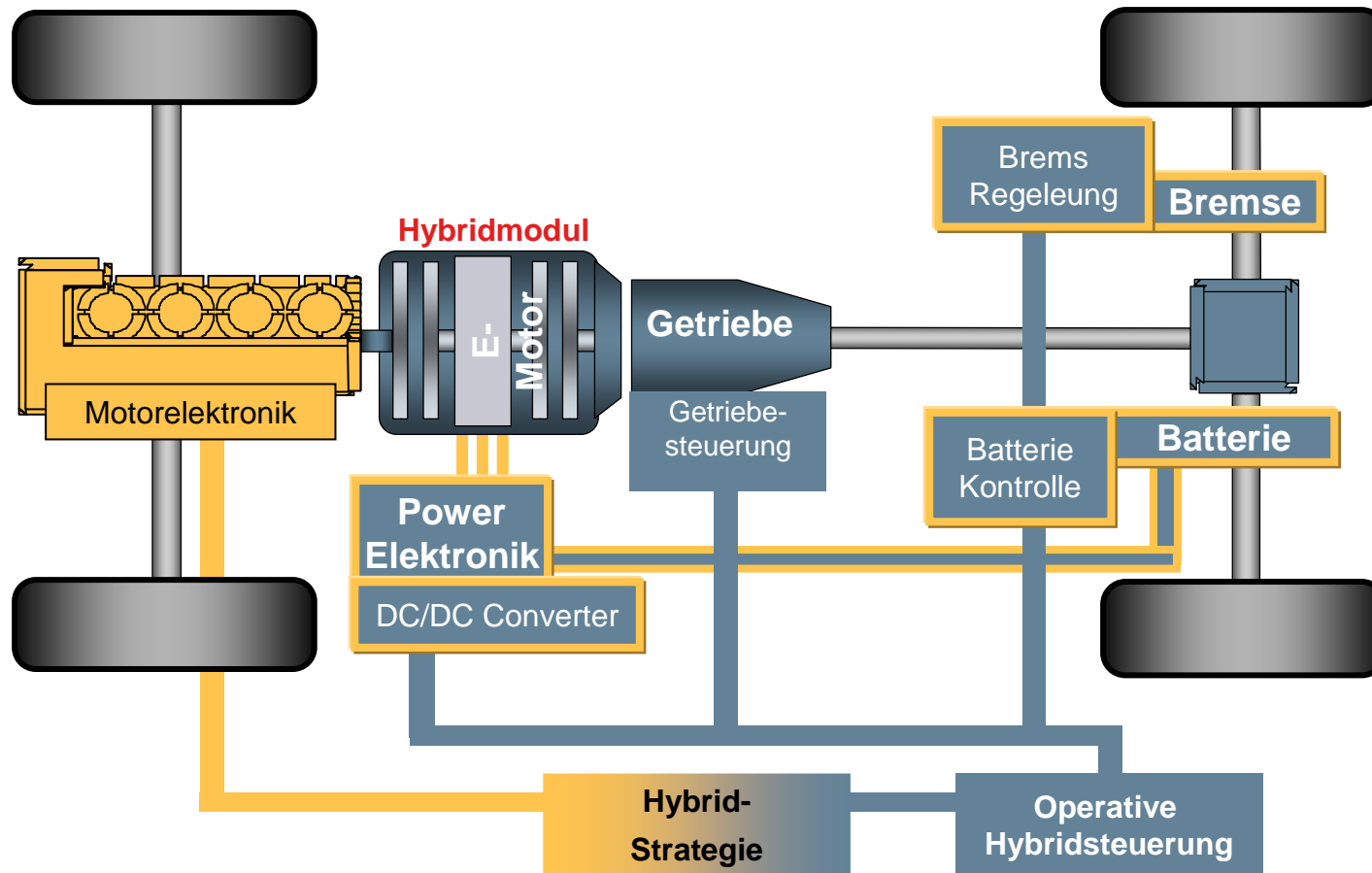
M. Neumann, M. Nass, C. Paulus (OTEH),
M. Tatar

ZF Friedrichshafen AG
QTronic GmbH





Hybridisierung des Antriebstrangs - Motivation





Hybridisierung des Antriebstrangs - Auswirkungen / Fragestellung

Auswirkungen:

Vielzahl
neuer Funktionen
im Antriebstrang

Anstieg funktionaler
Vernetzung von
Aggregaten

Integration neuer
Funktionen in vorh.
Architektur

Zentrale Fragestellungen:

Welche Entwicklungs- und Testmethoden sind notwendig,
um diese stark gestiegene Komplexität beherrschbar zu machen?

Wie kann die notwendige Testabdeckung trotz wachsendem
Funktionsumfang im gegebenen Zeitfenster sichergestellt werden?



Durchgängige Teststrategie - Zentrale Aspekte

- Verlagerung von Testumfängen von HiL-Simulator und Fahrversuch in VSiL.
- Automatisierte Testfallgenerierung, -Ausführung und -Bewertung.
- Systematische Modellierung von Funktions- und Softwareanforderungen in Form von Systeminvarianten.



PC-Systemintegration
ZF Softcar (SiL, VSiL)



Softwareabsicherung
HiL, VHiL

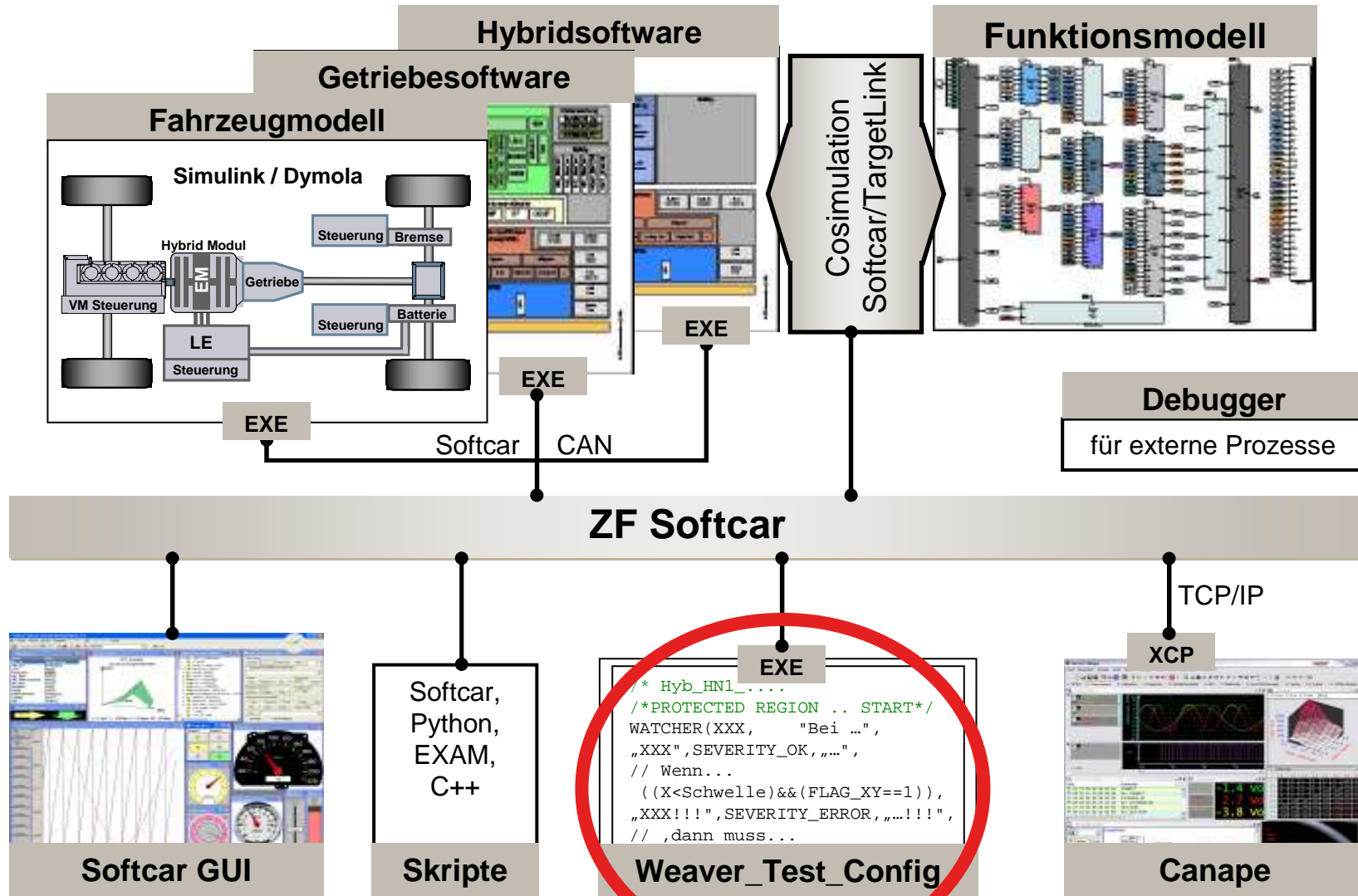


Systemtest
Mess-/Verstelltool (Fzg)



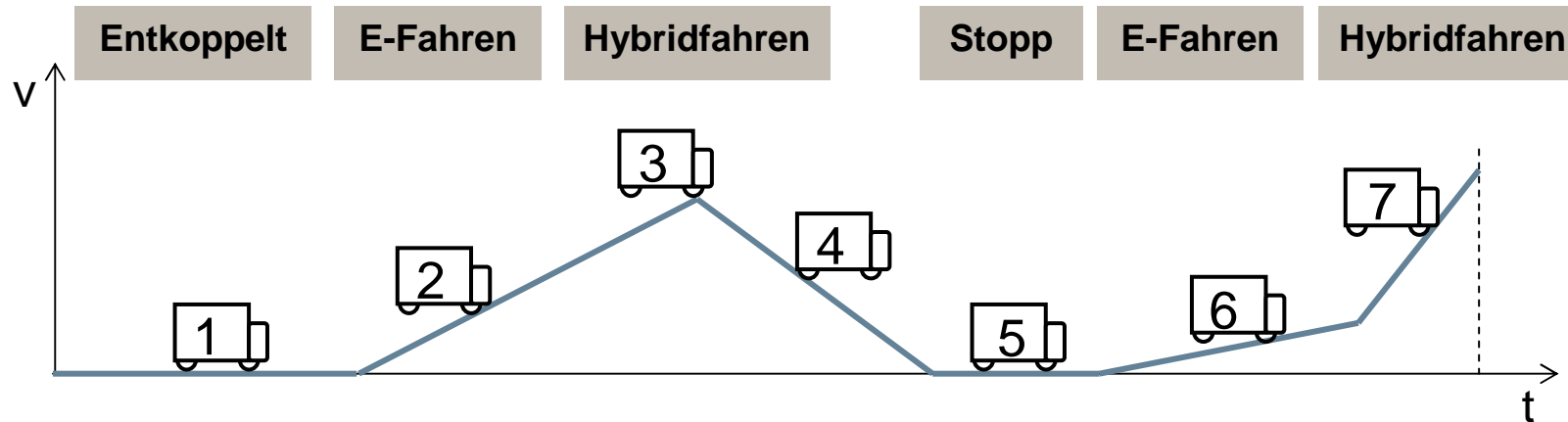
Software Prüfung mit Softcar-VSiL

Systemintegration Hybrid ...





Automatisierte Methode zur Testfallgenerierung - Motivation

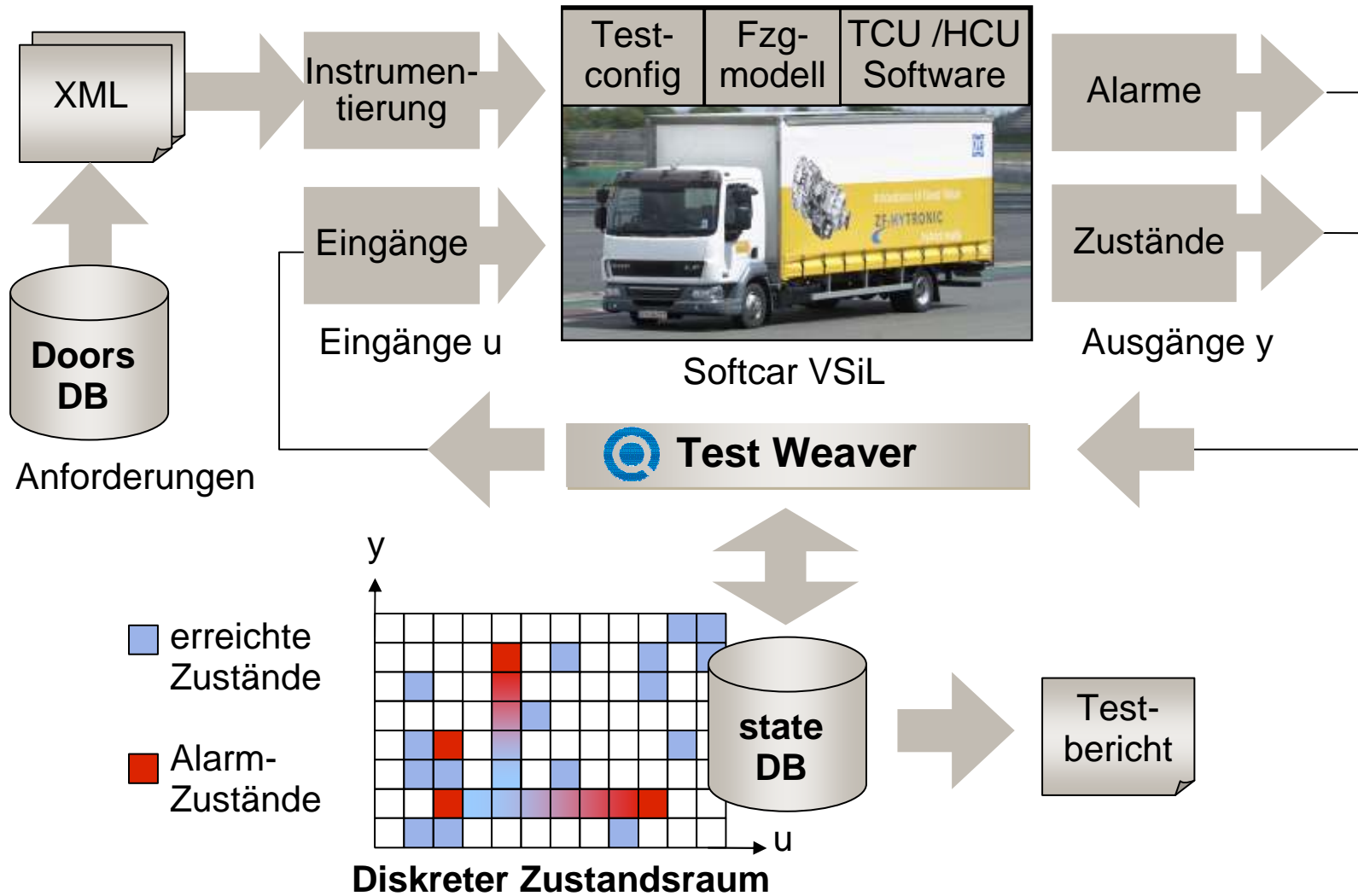


	Konventionelles Testskript
1	Anfangs SOC = 70%
2	Fahrschalter = D, Anfahren (Gas = 60%)
3	Bis 50 km/h beschleunigen
4	Bremsen bis zum Stillstand
5	Start-Stop wird aktiv
6	Moderat beschleunigen (Gas = 40%)
7	Kräftig beschleunigen (Gas = 90%)

Nachteile
Anforderungen werden nur in speziellen Fahrsituationen geprüft.
Negative Testbedingung sind schwierig zu formulieren.
Es sind viele unterschiedliche Tests notwendig, um eine hohe Testabdeckung zu erreichen.



Konfiguration VSiL Testumgebung mit Test Weaver





Konfiguration TestWeaver für Softcar VSiL

- Schnittstelle TestWeaver - Softcar

- ✓ Verbindung zwischen Softcar und TestWeaver über TCP/IP herstellen.
- ✓ Instrumentierung der Eingangsvariablen
- ✓ Instrumentierung der Ausgangsvariablen
- ✓ Implementierung der System- und Softwareanforderungen

```
EXE
/* Hyb_HN1_....
/*PROTECTED REGION .. START*/
WATCHER(XXX, "Bei ...",
„XXX“, SEVERITY_OK, „...“,
// Wenn...
((X<Schwelle)&&(FLAG_XY==1)),
„XXX!!!“, SEVERITY_ERROR, „...!!!“,
// ,dann muss...
```

Weaver_Test_Config



Konfiguration TestWeaver für Softcar VSiL

- Eingangsvariablen (Chooser)

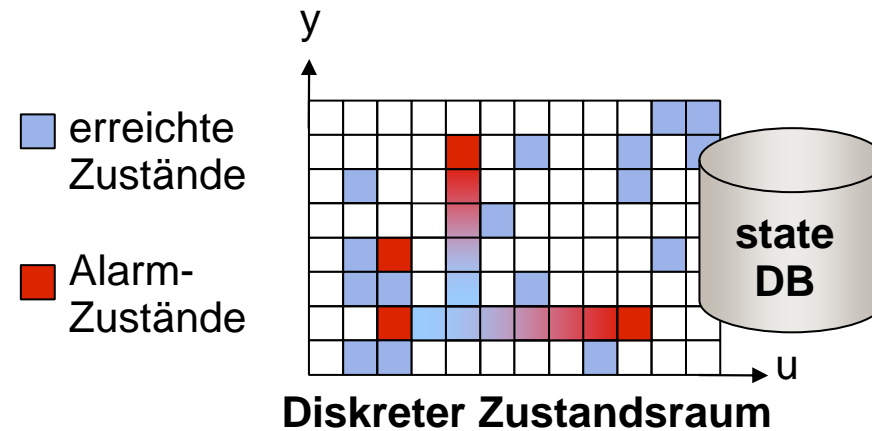
- Gaspedal, Bremspedal:
0%, 25%, 50%, 75%, 100%,
- Fahrschalter:
P, R, N, D,
- Parkbremse:
aus, ein,
- Starttemperatur Verbrennungsmotor:
kalt, warm,
- Fahrbahnsteigung:
-10% -5%, 0%, 5 % 10%,
- Anfangsladezustand der Hybridbatterie:
niedrig, mittel, voll

```
// Gas
PARTITION(Gaspedal) //Name des Choosers/Reporters
    {{{0,0}, "0%", OCCURRENCE_OK, "low speed"},
    {{25,25}, "25%", OCCURRENCE_OK, "low speed"},
    {{50,50}, "50%", OCCURRENCE_OK, "medium speed"},
    {{75,75}, "75%", OCCURRENCE_OK, "medium speed"},
    {{100,100}, "100%", OCCURRENCE_OK, "high speed"}}};
CHOOSE(Gaspedal, // Name des Choosers im TW
GasTW, // Softcar Variable
"%", // Einheit
"rel stellung des Gaspedals");// Beschreibung des Choosers
```



Konfiguration TestWeaver für Softcar VSiL

- Ausgangsvariablen (Reporter)



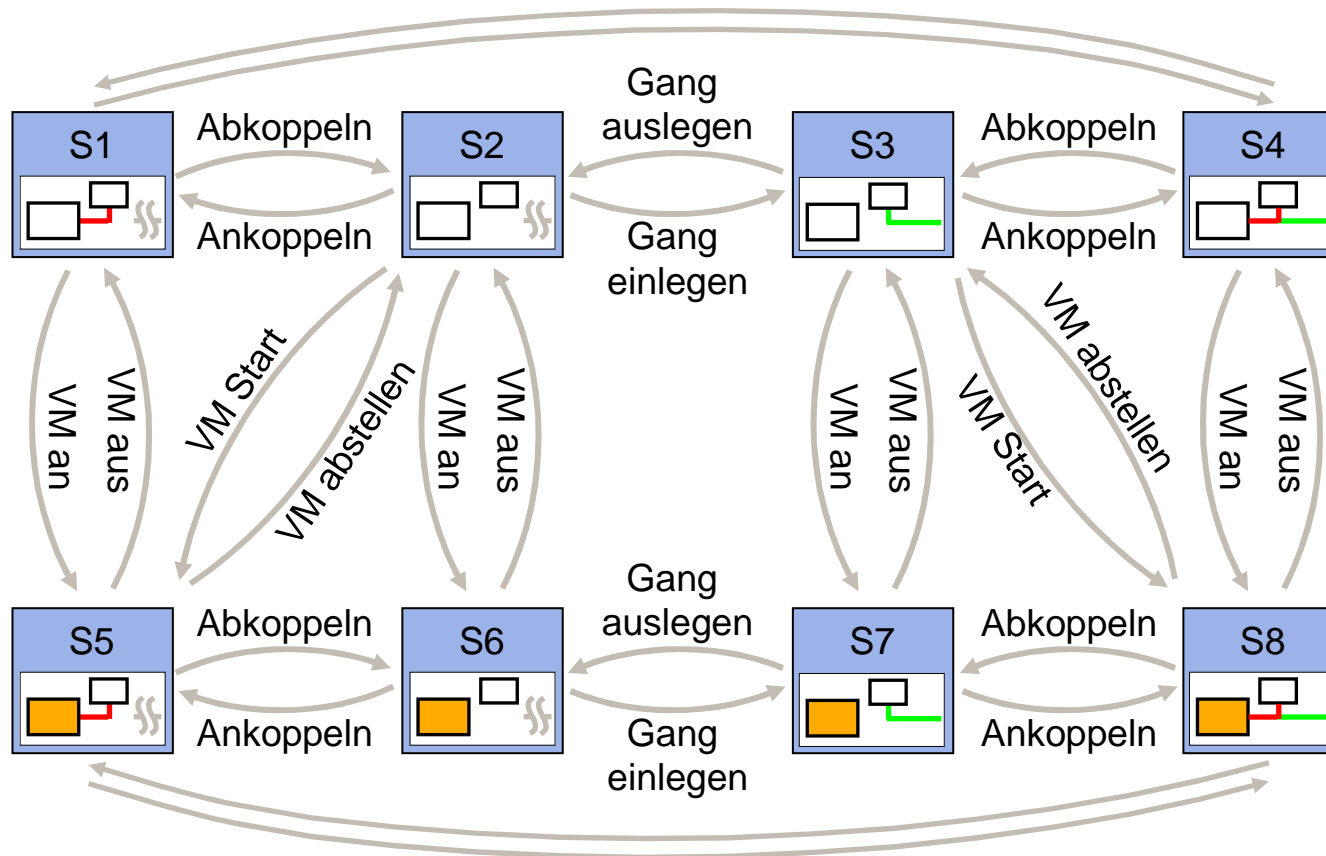
z.B.: Ladezustand der Hybridbatterie

```
PARTITION(SoC)
  {{0,20}, " 0..20%", SEVERITY_ERROR, ""},
  {{21,50}, "21..50%", SEVERITY_LOW, ""},
  {{51,80}, "51..80%", SEVERITY_OK, ""},
  {{81,90}, "81..90%", SEVERITY_HIGH, ""},
  {{91,100}, "91..100%", SEVERITY_ERROR, ""}};

REPORT(SoC,SOC,"%", "Ladezustand der Hybridbatterie");
```

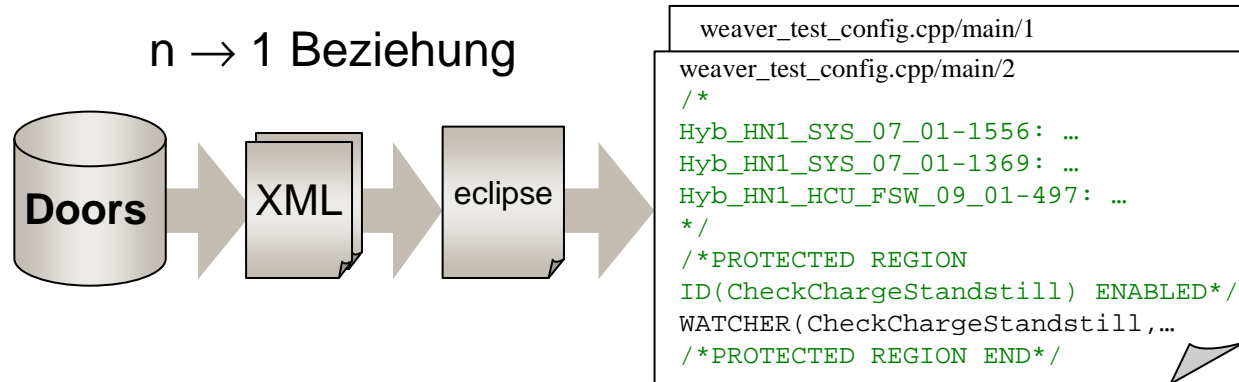


Automatisierte Methode zur Testfallgenerierung - 8 Antriebstrangzustände





Überprüfung der SW und Systemanforderungen - Traceability Doors - TestWeaver



"Immer wenn <Vorbedingung>, dann folgt <Nachbedingung> nach maximalem <Delay>".

Watcher-Instrumentierung

```
/* Hyb_HN1_SYS_07_01-1556: Bei einem niedrigen SOC ...
   Hyb_HN1_SYS_07_01-1369: Wenn die Energie im Speicher unterhalb ...
   Hyb_HN1_HCU_FSW_09_01-497: Wenn der SOC unter einer appllizierbaren Schwelle ...
/*PROTECTED REGION ID(CheckChargeStandstill) ENABLED START*/
WATCHER(CheckChargeStandstill, "Bei einem niedrigen SOC ...", "Wenn Energie < ...",
"ChargeStandstill,, SEVERITY_OK, "EM lädt die Batterie nicht!!!",
// Wenn ...
"ChargeStandstill!!!", SEVERITY_ERROR, "EM lädt die Batterie nicht!!!",
// dann muss ...
((HCU_SiEMot_M_trq_filt<=0) && (HCU_HSML_M_ctrlStrategicMode_req==4)), 0.5);
/*PROTECTED REGION END*/
```



Testergebnisse

- Coverage für Antriebstrangzustände und Operative Funktionen

Schaltungen

gear	gear_tgt	scenarios
R	R	[s0, s1]
	N	[s2, s1]
1	1	[s12, s92]
	R	[s0, s36]
N	N	[s0, s1]
	1	[s11, s12]
	2	[s11, s12]
	3	[s11, s12]
	4	[s11, s12]
	5	[s11, s12]
	6	[s17, s18]
1	R	[s119, s581]
	N	[s27, s28]
	1	[s11, s12]
2	2	[s11, s12]
	N	[s229, s230]
	1	[s201, s200]
3	2	[s11, s12]
	3	[s11, s12]
	N	[s374, s213]
	1	[s153, s152]
4	2	[s201, s200]
	3	[s11, s12]
	4	[s11, s12]
	N	[s518, s486]
	1	[s160, s168]
5	3	[s189, s336]
	4	[s11, s12]
	5	[s11, s12]
	N	[s166, s182]
	3	[s153, s152]
6	4	[s157, s141]
	5	[s11, s12]
	6	[s17, s18]
6	N	[s150, s325]
	5	[s144, s141]
	6	[s17, s18]

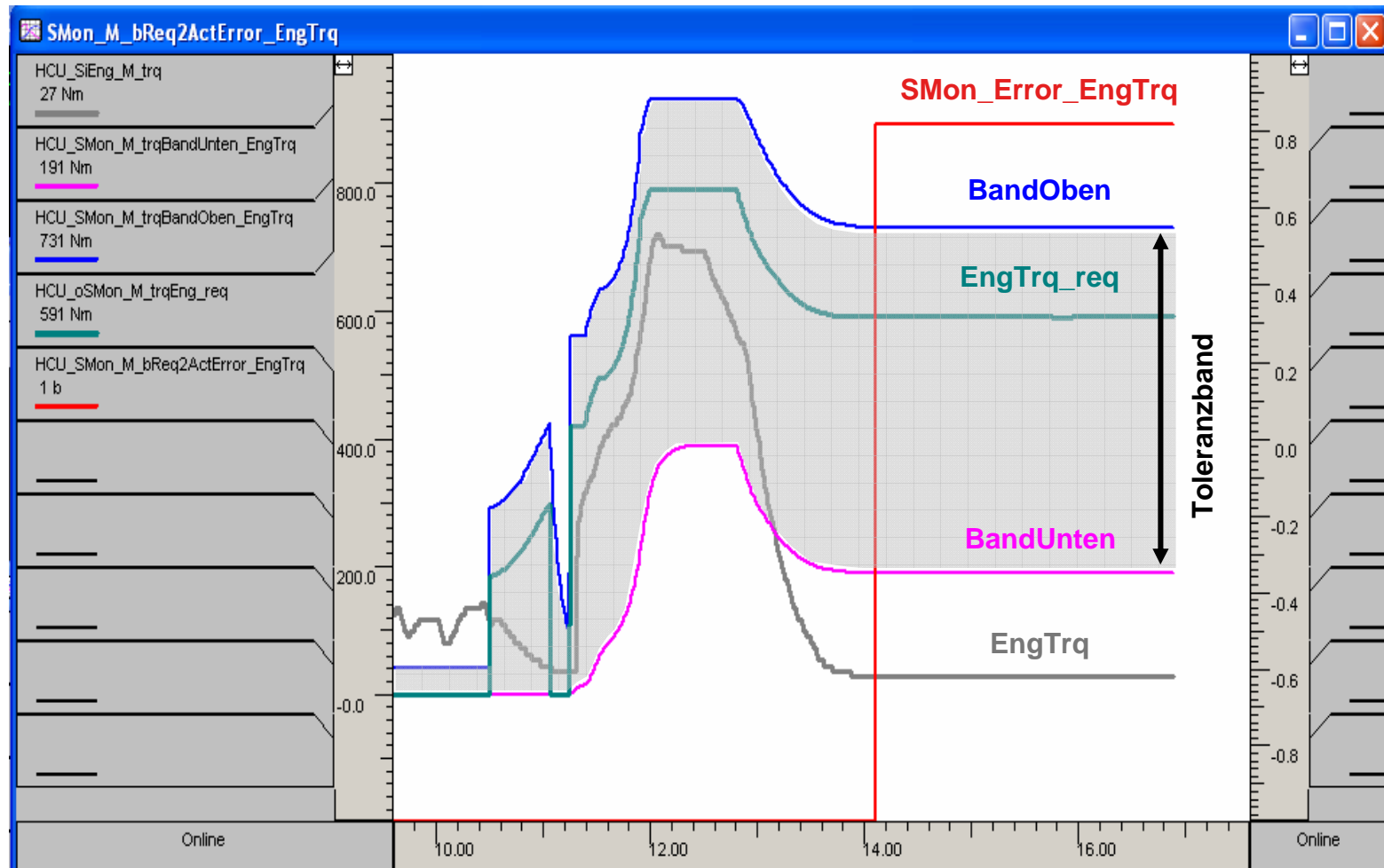
Antriebstrangzustände und operative Funktionen

Driveline Mode		Operating Mode	OUTPUT
current	requested	requested	scenarios
S0_WeaverDefault	S0_WeaverDefault	OpDefault	[s0, s49]
S1_Coupled	S1_Coupled	OpCharge	[s966, s993]
	S2_Decoupled	OpNeutralDecouple	[s966, s993]
	S8_HybridDrive	OpCharge	[s0, s1]
S2_Decoupled	S2_Decoupled	OpTransmissionCommand	[s361, s571]
		OpDrivelineOpen	[s990, s1000]
	S5_Charge	OpImpulseStart	[s974, s966]
	S6_DecoupledE	OpNeutralDecouple	[s974, s966]
...			...
...			...
...			...
S5_Charge	S6_DecoupledE	OpHybridDrive	[s361, s217]
	S7_EDriveEngOn	OpNeutralDecouple	[s0, s1]
	S8_HybridDrive	OpDrivelineOpen	[s0, s1]
S6_DecoupledE	S8_HybridDrive	OpTransmissionCommand	[s11, s12]
		OpCharge	[s0, s1]
		OpDrivelineOpen	[s0, s1]
S7_EDriveEngOn	S6_DecoupledE	OpDrivelineOpen	[s0, s1]
	S8_HybridDrive	OpTransmissionCommand	[s11, s0]
		OpDrivelineOpen	[s0, s1]
S8_HybridDrive	S7_EDriveEngOn	OpElectricDrive	[s677, s673]
	S8_HybridDrive	OpEnqCouple	[s992, s1070]
		OpTransmissionCommand	[s0, s1]
S8_HybridDrive		OpElectricDrive	[s0, s1]
		OpTransmissionCommand	[s0, s3]
		OpHybridDrive	[s4, s0]



Testergebnisse

- Beispiel: Fehlereintrag Safety-Monitor



Fehlercode 26 = SMon_ReqVal2ActValEmot: Fehler Soll/Ist-Vergleich E-Maschinen Moment/Drehzahl
Fehler aufgrund Unterschreitung Toleranzband



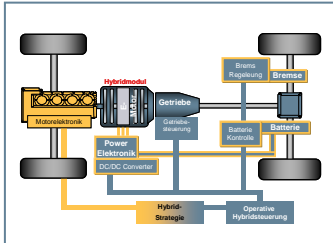
Testergebnisse

- Überwachungen

- ✓ Überwachung der Software- und Systemanforderungen,
- ✓ Überwachung des Fehlerspeichers (Funktions- und Safety-Layer),
- ✓ Überwachung von Wertebereichsüberschreitungen, Assertions, Resets, Prozessabstürze,
- ✓ Überwachung von „toggelnden“ Zustandsvariablen (Bits, Modi), etc..
- ✓ Ermittlung Code-Coverage mit CTC++

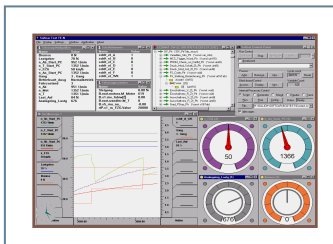


Zusammenfassung



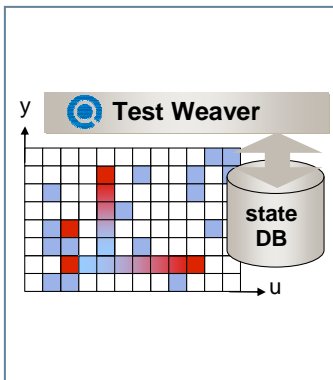
Hybridisierung von Antriebssträngen

- Neue Komponenten und Funktionen im Hybridsystem
- Steigende funktionale Vernetzung im Antriebstrang (hohe Komplexität)



Erweiterter Einsatz von Testwerkzeugen (VSiL) auf PC

- Frühe Absicherung von Teilfunktionalitäten auf PC im Systemverbund
- Co-Simulation ZF Softcar mit Test Weaver



Automatisierte Methode zur Testfallgenerierung

- Automatische Erzeugung, Simulation und Analyse tausender Szenarien
- Automatisches Erreichen und Überwachen vieler relevanter Systemzustände
- Hohe Testabdeckung, hoher Automatisierungsgrad,
- Vergleichbar geringer Spezifikationsaufwand.



Vielen Dank für Ihre Aufmerksamkeit!

