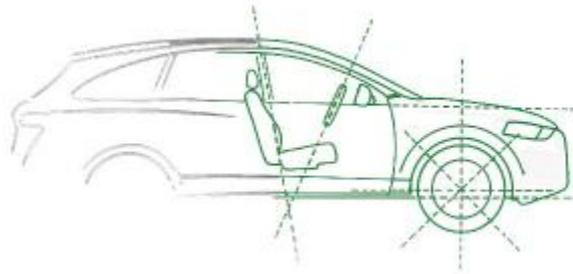




Durchgängiger Einsatz Hardware-unabhängiger Testfälle im MiL-, SiL- und HiL-Test

26. TAV Stuttgart



Michael Müller
Projektleiter
Berner & Mattner Systemtechnik GmbH
michael.mueller@berner-mattner.com

MM, 22.11.2007



Agenda

- Testverfahren in der Automobilindustrie
- Bestehende Probleme
- Lösung: Signalabstraktion
- Modell-Anbindung
- HW-Anbindung
- Testanbindung
- Einbindung in die Modellbasierte Entwicklung
- Diskussion

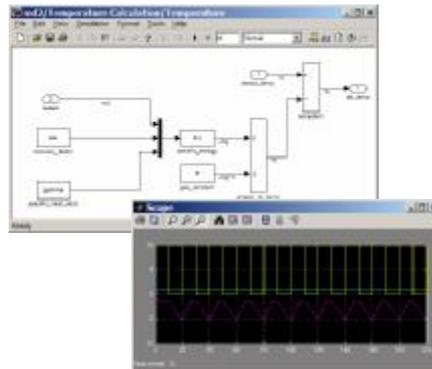
2



Testverfahren in der Automobilindustrie

MiL

- Getestet wird das Modell
- Eingangsgrößen werden simuliert
- Ausgangsgrößen werden protokolliert und können mit dem erwarteten Werten verglichen werden
- Automatisierbar über das verwendete Entwicklungswerkzeug oder Schnittstellen zu externen Tools



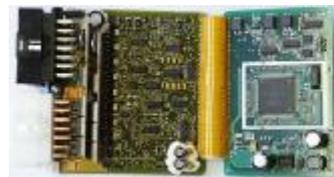
3



Testverfahren in der Automobilindustrie

SiL

- Getestet wird der generierte Code
- Simulation der Umgebung
- Verbindung der Ein- und Ausgänge mit dem Testsystem
- Der generierte Code wird auf dem PC oder einem Eval-Board ausgeführt
- Automatisierbar über das verwendete Entwicklungswerkzeug oder Schnittstellen zu externen Tools



4



Testverfahren in der Automobilindustrie

HiL

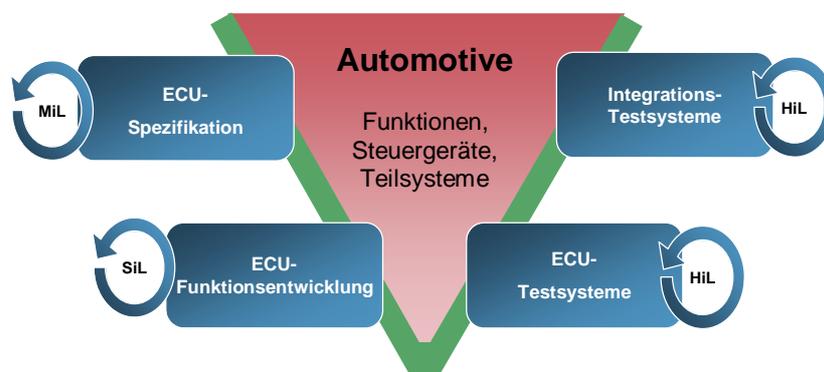
- Getestet wird das reale Steuergerätes
- Simulation der Umgebung durch Umgebungsmodelle (z.B. Matlab/Simulink)
- Verbindung der Ein- und Ausgänge mit dem HiL-Simulator
- Stimuli über den HiL-Simulator
- Vergleich der Ausgangswerte mit Erwartungswerten
- Automatisierbar über die Steuerungssoftware des HiL-Simulators



5



Testverfahren in der Automobilindustrie



6



Bestehende Probleme

- Testfälle aus früheren Entwicklungsphasen können in späteren Phasen nicht wiederverwendet werden (MiL-Testfälle können nicht für den SiL- bzw. HiL-Test verwendet werden)
- Steuergerätestests können erst erstellt werden, wenn das Steuergerät verfügbar ist
- Hohe Auslastung der HiL-Simulatoren durch das Erstellen der Tests
- Testfälle müssen bei Änderung der verwendeten HW neu erstellt bzw. angepasst werden
- Keine Wiederverwendung von Modellen in späteren Entwicklungsphasen
- Keine Verifikation der Steuergeräte gegen das Spezifikationsmodell

7



Lösung: Signalabstraktion

- Der Signalpool enthält alle im System verwendeten Signale
- Jedes Signal hat
 - einen Namen
 - einen Datentyp
 - eine Länge
 - eine Signal-ID
- Signale werden auf Basis einer Konfiguration erzeugt
- Alle verwendeten Komponenten kommunizieren über den Signalpool

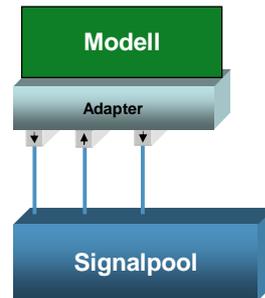


8



Modellanbindung

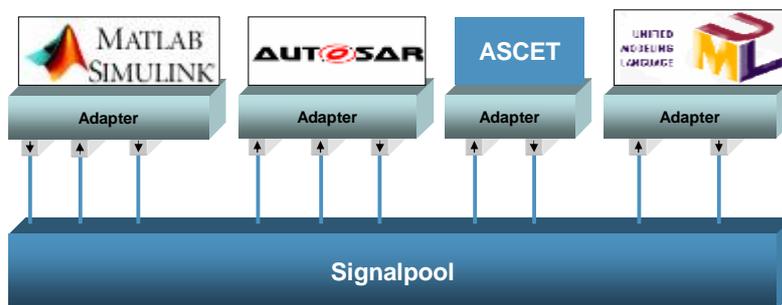
- Modelle greifen über Ports auf den Signalpool zu
- Die Verbindung Port zu Signalpool-Signal ist konfigurierbar
- Adapter können sich auf Signalpooländerung registrieren
- Adapter informieren bei Signalpooländerung das Modell
- Modelle haben keinerlei Informationen über:
 - Verwendete HW
 - Andere Kommunikationspartner
 - Test-Plattform



9



Kopplung verschiedener Modelltypen

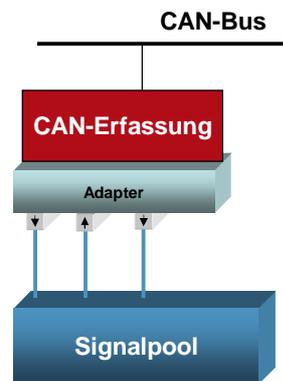


10



HW-Anbindung (Beispiel CAN)

- CAN-Signale werden auf Ports abgebildet
- Die Verbindung Ports zu Signalpool ist konfigurierbar
- Adapter registriert sich auf Signalpooländerung
- Adapter sendet bei Änderung automatisch die entsprechende CAN-Botschaft
- Eingehende CAN-Botschaften werden zerlegt und die entsprechenden Signalwerte im Signalpool abgelegt



11



Testanbindung

- Testfall greift nur auf den Signalpool zu
- Signale
 - WiperPoti
 - WiperLever
 - WiperStatus
- Testfall ist wiederverwendbar durch Parametrisierung
- Parameter
 - INTERVAL
 - TIME

```

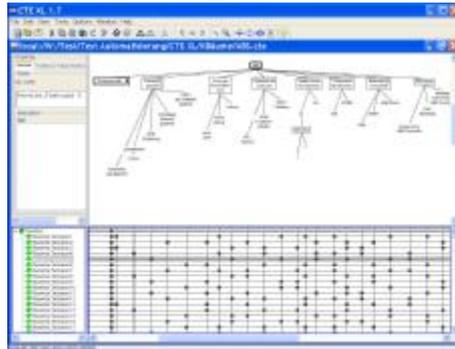
interval_wiping() {
  // set wiping interval
  WiperPoti.sendValue(INTERVAL);
  // set wiperlevel
  WiperLever.sendValue(1);
  // wait for wiper to start
  ASSERT(WiperStatus.wait(true, 1000));
  // wait for wiper to stop
  ASSERT(WiperStatus.wait(false, 2000));
  // wait for 2nd interval
  ASSERT(WiperStatus.wait(true, TIME));
  // wait for wiper to stop
  ASSERT(WiperStatus.wait(false, 2000));
  // switch off wiper
  WiperLever.sendValue(0);
}
  
```

12



Variantenbildung mit der Classification Tree Methode

- Definition der Parameter (evtl. abgeleitet aus den Testfällen)
- Festlegung der möglichen Parameterwerte
- Konfiguration der Varianten
- Generierung der Varianten
- Definition der Sollwerte
- Verwendung zur Parametrisierung der Testfälle



13



Einbindung in die Modellbasierte Entwicklung

Automobil-Hersteller...

- erstellt Spezifikationsmodelle
- erstellt Testfälle
- testet Modelle einzeln oder im Verbund mit
 - anderen Modellen
 - Generiertem Code
 - Steuergeräten
- gibt Modelle und Testfälle an Zulieferer

14



Einbindung in die Modellbasierte Entwicklung

Automobil-Zulieferer...

- verfeinert Modelle
- generiert Code
- erstellt Testfälle
- testet Code einzeln oder im Verbund mit
 - Modellen
 - anderem generierten Code
 - Steuergeräten
- erstellt das Steuergerät
- testet Steuergerät einzeln oder im Verbund mit
 - Modellen
 - generiertem Code
 - anderen Steuergeräten
- gibt Steuergerät und Testfälle an Hersteller

15



Einbindung in die Modellbasierte Entwicklung

Automobil-Hersteller...

- testet Steuergerät einzeln oder im Verbund mit
 - generiertem Code
 - Spezifikationsmodellen
 - anderen Steuergeräten

16



Zusammenfassung

- Testfälle sind unabhängig von
 - der verwendeten Umgebung
 - der angebundenen HW
- Testfälle können parametrisiert werden
- Testfälle können bereits in frühen Entwicklungsphasen erstellt werden und in späteren Phasen wiederverwendet werden
- Testfälle lassen sich aus den verwendeten Modellen ableiten
- Modelle können zur Verifikation der Steuergeräte verwendet werden

17



Vielen Dank

Noch Fragen?

18