

Mit der Verbreitung von AUTOSAR wird auch der effiziente Test von AUTOSAR-Komponenten mit maßgeschneiderten Testlösungen immer wichtiger. Im Beitrag wird dargestellt, wie mit TPT, dem modellbasierten Testwerkzeug der Firma PikeTec, AUTOSAR-Komponenten einzeln und im Verbund getestet werden. Damit können bereits für Modelle erstellte Testfälle für SiL und HiL weiterverwendet werden. Diese Durchgängigkeit beim Test vereinfacht das Testen und steigert die Effizienz der Verifikation.

Einleitung

AUTOSAR, das ist die AUTomotive Open System ARchitecture, ein Standard für die Entwicklung von Steuergerätesoftware, der aus einem Zusammenschluss mehrerer Automobilhersteller und Zulieferern entsteht [1]. Ziel ist es die Softwareentwicklung von der Hardwareentwicklung zu entkoppeln und einen Standard zu schaffen, der es ermöglicht Software auf verschiedener Hardware laufen zu lassen.

In AUTOSAR kommunizieren Softwarekomponenten über wohldefinierte und standardisierte Schnittstellen, sogenannte Ports und über einen Virtual Functional Bus (VFB), der die hardwarenahe Kommunikationsschicht auf dem Steuergerät sowie die Kommunikation zwischen mehreren Steuergeräten kapselt. Komponenten können dadurch auf immer gleiche Weise miteinander kommunizieren, unabhängig davon auf welchen Steuergeräten sie sich befinden. Aus dem Mapping der Komponenten auf Steuergeräte und der Netzwerktopologie kann für jedes Steuergerät eine so genannte RTE generiert werden, welche die notwendige Kommunikation implementiert [2].

AUTOSAR Softwarekomponenten stellen die eigentliche Funktionssoftware dar, die jeweils Steuerungs- und Regelungsfunktionen ausführt. Die Entwicklung der Softwarekomponenten erfolgt in der Regel zunächst unabhängig von der Implementierung auf einem oder mehreren Steuergeräten. Erst in späteren Entwicklungsschritten erfolgt eine Verteilung der Software auf die Steuergeräte.

AUTOSAR Softwarekomponenten werden funktional wie andere Steuerungen entwickelt. Sie werden in C implementiert oder modellbasiert beispielsweise mit MATLAB/Simulink und TargetLink oder ASCET entwickelt. Im Fall der modellbasierten Entwicklung wird aus den Modellen automatisch C-Code generiert.

AUTOSAR Authoring Tools sind die Werkzeuge, die die Softwarekomponenten und ihre Schnittstellen als AUTOSAR-Komponenten beschreiben, auf die einzelnen ECUs verteilen und die RTEs der ECUs generieren. Dabei wird die sogenannte Softwarekomponenten Beschreibungsdatei, eine ARXML-Datei mit einer Schnittstellenbeschreibung, den Operationen der Komponenten und ihren Schnittstellen erzeugt.

Für die Simulation und den Test von AUTOSAR-Komponenten stehen heute bereits einige Werkzeuge verschiedener Hersteller bereit. Dazu zählen SystemDesk von dSPACE, DaVinci Component Tester von Vektor Informatik oder Messina von Berner und Mattner.

Alle Werkzeuge simulieren einen VFB und können AUTOSAR-Software-Komponenten ansprechen und ausführen. Die Stimulation der Software erfolgt über werkzeugspezifische Schnittstellen auf unterschiedliche technische Weise und meist nur mit einfachen Mitteln. Für umfangreiche

wohlstrukturierte und wiederverwendbare Tests, die das funktionale Verhalten möglichst umfassend abbilden, reicht das nicht aus. Stattdessen kann hierfür alternativ das modellbasierte Testwerkzeug TPT verwendet werden [8].

TPT

TPT ist ein modellbasiertes Testwerkzeug, das sich vornehmlich für den Test von Steuerungs- und Regelungssystemen eignet [3]. AUTOSAR-Komponenten gehören zu dieser Systemklasse.

Der Testprozess mit TPT läuft in vier Schritten ab. Die manuelle Testmodellierung geht den automatisierten Prozessschritten Testausführung (Execution), Testauswertung (Assessment) und Testdokumentation (Reporting) voran (vgl. Abbildung 1).

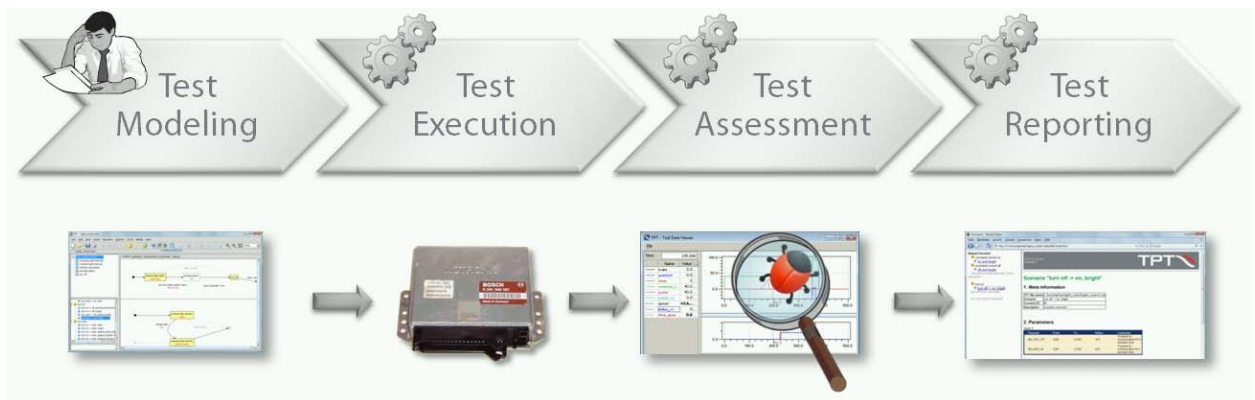


Abbildung 1: Testprozess mit TPT

Mit TPT werden funktionale Testfälle mittels einfach zu verstehenden Ablaufautomaten oder Sequenzbeschreibungen modelliert. Das besondere an diesen Automaten ist, dass sie reaktiv sind. Es besteht also die Möglichkeit, den Testablauf von Messungen am zu testenden System abhängig zu machen. So kann beispielsweise beim Test einer Getriebesteuerung auf das Eintreten eines Gangwechsels oder das Erreichen einer bestimmten Geschwindigkeit (Vgl. Abbildung 2) im Testablauf gewartet werden oder es kann eine Geschwindigkeitsregelung realisiert werden.

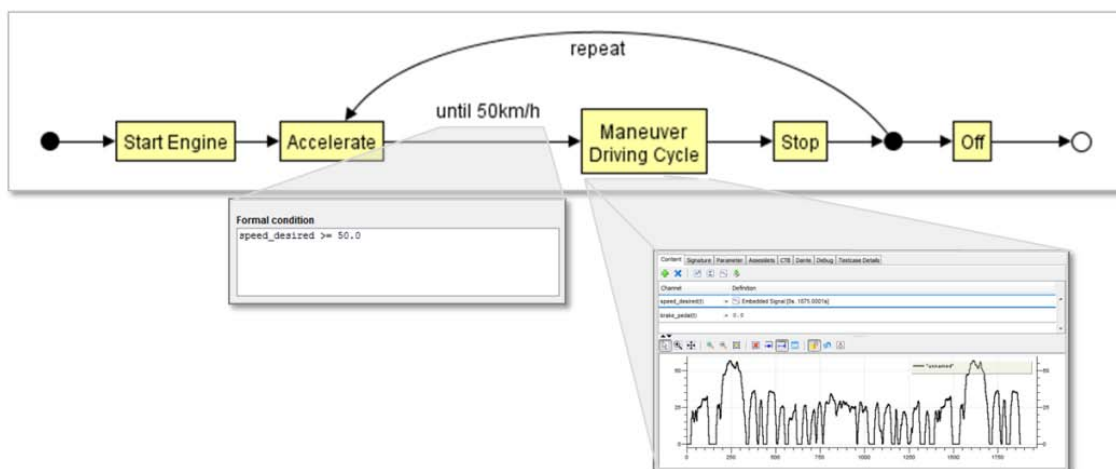


Abbildung 2: Hybride Ablaufautomaten

Der Testablauf muss also nicht vollständig a priori beschrieben sein, sondern kann sich abhängig vom Systemverhalten ändern. Eine weitere Eigenschaft der Automaten in TPT ist, dass die Gesamtheit aller Testfälle für ein System mittels eines Automaten modelliert werden kann. Dies wird erreicht durch Varianten von Zuständen. Durch Variation und Kombination der Varianten entstehen Testfälle (Vgl. Abbildung 3), die zudem systematisch entwickelt werden können, weil sie sich die Vorgehensweise auf die Äquivalenzklassen- oder Klassifikationsbaummethode [4] abbilden lässt.

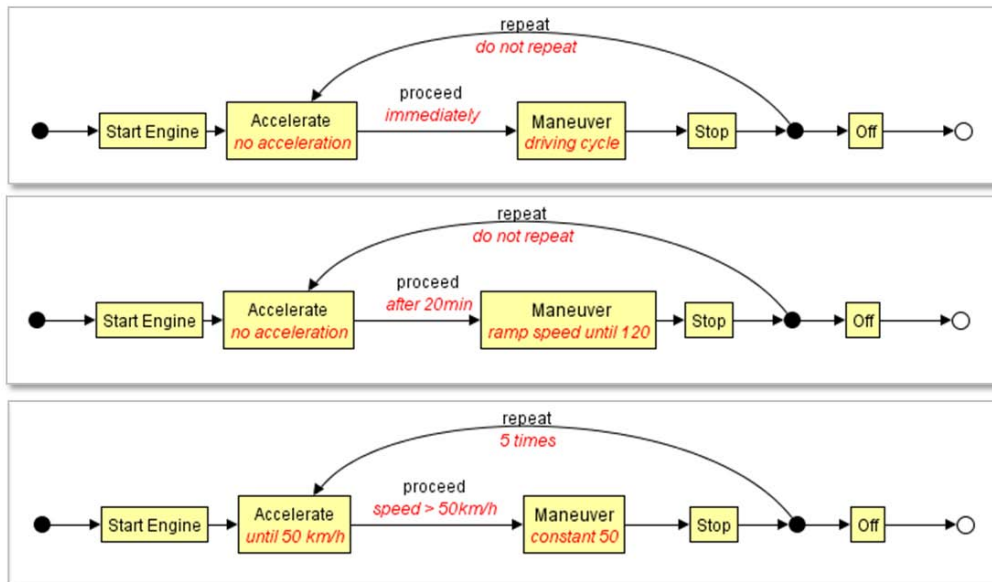


Abbildung 3: Drei unterschiedliche Manöver (Kombination von Varianten)

Die Automaten sind hierarchisch und können parallel ablaufen. Auf der oberen Ebene sind für den Nutzer Notationen in Klartext sichtbar, die auch Nicht-Programmierern das Verständnis der Testfälle ermöglichen. Auf der untersten Ebene der Automaten verbirgt sich immer eine Beschreibung des Verhaltens als Funktion der Zeit der Form $y(t) = f(x, u, y)$, wobei y der Signalausgang, u der Eingang und x ein anderes Signal des Modells darstellt. Damit sind konstante Signalverläufe aber auch komplexe Funktionen beschreibbar. Signale können auch flexibel mittels eines Signaleditors erzeugt oder Messsignale importiert werden.

Eine alternative Testbeschreibung, die sich nahtlos und vollständig in die Automatenbeschreibung einbettet, ist die sogenannte „Test Step List“. Mit dieser einfachen Methode lassen sich mittels Sequenzen einfacher Befehle einfache Stimulationen einfach realisieren.

1	Set channel	brake_pedal	:= 0		
2	Ramp channel	acceleration_pedal	to 80	with gradient 10	per sec
3	Set channel	ignition	:= 1		
4	Wait for value	speed_measured	== 55	with tolerance +/-0	
5	Set channel	brake_pedal	:= 20		
6	Wait	5s			

Abbildung 4: Test Step List

TPT-Testfälle sind auf einer großen Bandbreite von Ausführungsplattformen wie beispielsweise MATLAB/Simulink, ASCET, C-Code, SystemDesk, DaVinci und MESSINA aber auch via CAN, LIN oder in Echtzeitumgebungen ausführbar. Die Ausführung auf verschiedensten Plattformen ist möglich, weil die Testausführung auf einer hochoptimierten Test-Maschine basiert, die in ANSI-C implementiert ist und modular in unterschiedliche Testumgebungen auf Host- und Targetsystemen integriert werden kann. Diese so genannte Virtuelle Maschine (TPT-VM) berechnet die mit TPT erstellten Tests und kann problemlos auf fast jedem System

implementiert werden. Auf Echtzeit-Testplattformen ist auch die Durchführung von TPT-Testfällen in Echtzeit garantiert.

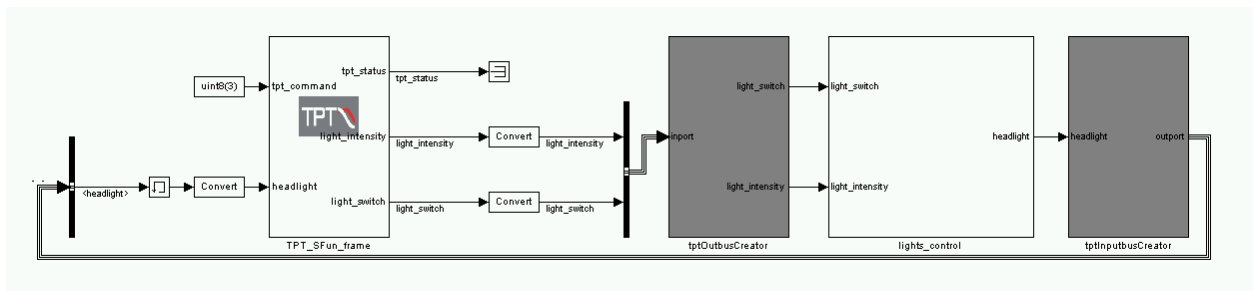


Abbildung 5: Generierter Testtreiber in MATLAB/Simulink

Die automatisierte Testauswertung kann mit TPT online, offline oder in einer Kombination aus Online- und Offline-Auswertung erfolgen. Bei der Online-Testauswertung erfolgt die Berechnung während der Testausführung und bei der Offline-Testauswertung nach der Testausführung. Der Vorteil der Online-Auswertung ist, dass unmittelbar auf erkannte Fehler reagiert werden kann (z.B. durch Testabbruch).

Die Offline-Auswertung, auch Assessment genannt, ermöglicht es, mit anderen Messwerkzeugen „extern“ aufgezeichnete Signale in die Auswertung einzubeziehen. Für die Auswertung steht eine umfangreiche Auswertesprache zur Verfügung. Diese lässt nicht nur einfache Abfragen auf das Erreichen eines bestimmten Ausgangswertes zu, sondern hält viele komplexe Signalauswertungen bereit wie beispielsweise Signalvergleiche mit Zeit und Wertetoleranz, Minimum- und Maximum-Vergleich, Sprungerkennung oder Filteroperationen (IIR, FIR). Mittels zeitlicher regulärer Ausdrücke können für die Testauswertung interessante Passagen aus der Fülle von Messdaten ausgeschnitten und ausgewertet werden. So ist es beispielsweise

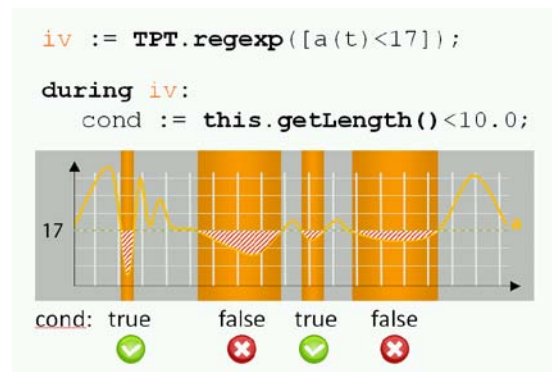


Abbildung 6: Beispielassessment

einfach möglich den Bereich zu suchen, in dem zwei Gangwechsel in einem Getriebe in Folge auftreten.

Die Ergebnisse der Tests und der automatischen Testauswertung können in einem Signal-Browser dargestellt werden. Zusätzlich stehen frei konfigurierbare Reports zur Verfügung, die einerseits die Testergebnisse enthalten, aber auch Assessment-Informationen. Der Testreport enthält detaillierte Informationen zum Testergebnis. Im Testreport selbst kann auch interaktiv ggf. nach Fehlerursachen geforscht werden kann. Die Darstellung erfolgt übersichtlich. Eine manuelle Verifikation der Testergebnisse entfällt.

Tests von Modellen

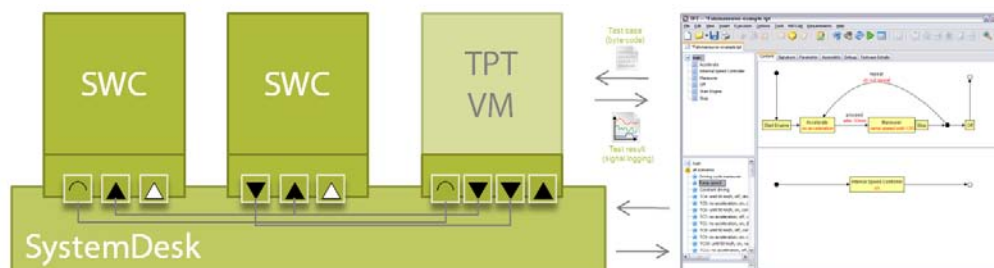
Während der Entwicklung von AUTOSAR-Komponenten steht bei der modellbasierten Entwicklung erstmals eine ausführbare Softwarekomponente nach der Modellierung zur Verfügung. Diese Modelle, die beispielsweise in MATLAB/Simulink, TargetLink oder ASCET implementiert sind, können mit TPT getestet werden. Dafür können die Schnittstellen der Modelle (MATLAB/Simulink, TargetLink, ASCET) automatisch analysiert und Testrahmen in den Modellierungswerkzeugen erzeugt werden. Auf dieser Basis können die Modelle dadurch systematisch, effektiv und komfortabel schon in sehr frühen Entwicklungsphasen mit TPT getestet werden.

Wird aus den Modellen im weiteren Verlauf des Entwicklungsprozesses C-Code generiert oder wird der Code manuell erstellt, können die Softwarekomponenten als AUTOSAR-Komponenten in SystemDesk, DaVinci CT oder Messina zur Verfügung gestellt und dort mit den in TPT erstellten Testmodellen einzeln oder integriert getestet werden. Aufgrund der Plattformunabhängigkeit der Testmodelle von TPT können die Tests dabei in der Regel 1:1 wiederverwendet werden. Nur das Setup des Testrahmens unterscheidet sich je nach AUTOSAR Werkzeug. Die notwendigen Schritte und Besonderheiten werden in den folgenden Abschnitten dargestellt.

Test von AUTOSAR Komponenten mit TPT und SystemDesk

Mit SystemDesk [5] können AUTOSAR-Softwarearchitekturen und das Mapping auf Steuergeräte und die Netzwerkkommunikation entwickelt werden. Darüber hinaus ist mit SystemDesk die Simulation einzelner oder mehrerer Softwarekomponenten möglich.

Für die Testautomatisierung in SystemDesk mit TPT wird die Laufzeitumgebung von TPT, die TPT-VM, als AUTOSAR-Komponente mit in die Konfiguration eingebunden. Die TPT-VM kann hierbei auf einem zusätzlichen virtuellen Steuergerät (Test-ECU) oder auf einem existierenden virtuellen Steuergerät integriert werden. Die Komposition besteht dann aus der TPT-VM und der zu testenden Softwarekomponente (SWC). Die TPT-VM stellt alle erforderlichen Signale für den Test auf dem RTE bereit und liest für die Testauswertung (Assessments) erforderliche Signale. Um die TPT-VM für die RTE-Generierung mit den relevanten Ports zu konfigurieren, wird die Interface-Beschreibung aus den SW-Component-Description-Files automatisch in TPT eingelesen und daraus die Softwarekomponente für die TPT-VM mit den passenden Schnittstellen erzeugt. Die Anbindung erfolgt dadurch nahezu automatisch.



Mit TPT können sowohl einzelne Softwarekomponenten als auch der Verbund verschiedener Softwarekomponenten – auch über ECU-Grenzen hinweg – mit SystemDesk getestet werden. Wenn die Simulationsumgebung einmal konfiguriert und die RTE generiert ist, können alle Testfälle automatisiert durchgeführt werden. Die gesamte Simulations- und Teststeuerung übernimmt TPT, indem TPT SystemDesk fernsteuert. Dadurch sind auch Batchtests mit vielen Testfällen vollautomatisch möglich.

AUTOSAR Komponenten mit DaVinci testen

DaVinci Component Tester (kurz: DaVinci CT) [6] stellt eine .NET-Infrastruktur bereit, die es ermöglicht AUTOSAR-Softwarekomponenten oder auch den Verbund (Komposition) mehrerer Softwarekomponenten zu testen. Dafür implementiert DaVinci CT den Virtual Functional Bus (VFB) und kann diesen über eine einheitliche .NET-API simulieren. Der VFB ist die Implementierung der RTE für einen PC. An den VFB können die Softwarekomponenten angeschlossen und simuliert werden. Im Unterschied zu SystemDesk werden keine Buskommunikation, Scheduling, Events oder Interrupts simuliert. Für das Testen der Softwarekomponenten werden die Interfaces durch DaVinci CT als .NET-API

bereitgestellt. Die Aufrufe der Runnables aus den Softwarekomponenten erfolgt aus TPT. Hierfür wird die TPT-VM mit dem .NET-API von DaVinci CT damit mit der Softwarekomponente verbunden. Diese Integration erfolgt automatisiert. TPT liest die Interface-Beschreibung aus den SW-Component-Description-Files automatisch in TPT ein und erzeugt daraus der Testrahmen für die Verknüpfung der TPT-VM mit der .NET-API mit den passenden Schnittstellen. Auch hier kann der Test vollautomatisiert von TPT durchgeführt werden.

Testing AUTOSAR Components with MESSINA

MESSINA heißt das Testwerkzeug von Berner und Mattner [7].

Bei MESSINA erfolgt die Stimulation Softwarekomponenten über einen sogenannten Signalpool. D.h. die zu testende AUTOSAR-Komponente ist in MESSINA eingebunden. Das Ansprechen der Software erfolgt über ein Signal-Mapping. Der Test mit MESSINA ist durchgängig von Modelltest bis zum HiL einsetzbar. Für den Test mit TPT bedeutet das, dass die Laufzeitumgebung (TPT-VM) auf den Signalpool zugreifen kann und somit auf alle Signale der Softwarekomponente zugreifen kann. Möglich ist bei MESSINA auch die Verbindung mit Umgebungsmodellen.

Fazit

Das Softwaretestwerkzeug TPT der Firma PikeTec kann für den Test von AUTOSAR-Softwarekomponenten eingesetzt werden. Um das zu ermöglichen kann TPT eine Reihe von Softwarewerkzeugen, die in der AUTOSAR-Entwicklung eine Rolle spielen, fernsteuern. Neben dem Test in frühen Entwicklungsphasen von MATLAB/Simulink-, TargetLink- oder ASCET-Modellen ist der Test von AUTOSAR Softwarekomponenten mittels DaVinci CT, SystemDesk oder MESSINA möglich. In allen Fällen können eine oder mehrere Softwarekomponenten getestet werden. Durch die Verwendung eines einheitlichen Testwerkzeugs von Model-in-the-Loop über Software-in-the-Loop bis hin zu Hardware-in-the-Loop können einmal erstellte Testfälle zu großen Teilen wiederverwendet werden. Darüber hinaus ist die Dokumentation der Tests einheitlich. Mit der durchgängigen Verwendung ein und derselben Testlösung können der Testprozess und die Qualität der Softwareprodukte bei gleichzeitiger Zeiteinsparung verbessert werden.

Literatur

[1] URL: <http://www.autosar.org/> ; Datum: 29.3.2010

[2] URL: <http://www.autosar.org/index.php?p=3&up=1&uup=1&uuup=0> ; Datum: 29.3.2010

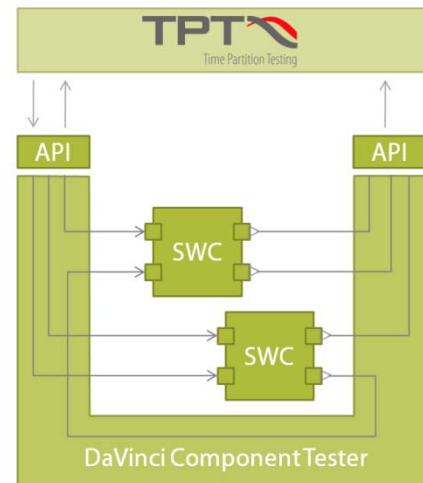
[3] Eckard Bringmann, Andreas Krämer. *Model-Based Testing of Automotive Systems*, In: [ICST, pp.485-493, 2008 International Conference on Software Testing, Verification, and Validation, 2008](#)

[4] Klaus Grimm: *Systematisches Testen von Software: Eine neue Methode und eine effektive Teststrategie*, München/Wien/Oldenburger 1995, GMD-Berichte Nr. 251.

[5] URL:

http://www.dspace.de/ww/de/gmb/home/products/sw/system_architecture_software/systemdesk.cfm ; Datum: 29.3.2010

[6] URL: http://www.vector.com/vi_davinci_component_tester_de,2816.html ; Datum: 29.3.2010



[7] URL: <http://www.berner-mattner.com/de/berner-mattner-home/produkte/messina/index.html> ;

Datum: 29.3.2010

[8] URL: <http://www.piketec.com/products/tpt.php> ; Datum: 29.3.2010

Kontakt

Dr. Jens Lüdemann

PikeTec GmbH

Waldenserstr 2-4

10551 Berlin

Tel: +49 30 394096830

mail: info@piketec.com

www: <http://www.piketec.com>