

# Modelle werden kompatibel

## Werkzeug zur Simulation mechatronischer Fahrzeugsysteme

Für die Modellierung mechatronischer Applikationen im Automotive-Umfeld gibt es mehrere Alternativen auf dem Markt. Diese sind zumeist proprietäre Lösungen, die für sich allein stehend sehr gut funktionieren. Die Probleme beginnen, wenn es um Schnittstellen oder Konsolidierung von Simulationsmodellen aus verschiedenen Tool-Umgebungen geht.



Von Thorsten Gerke und Ewald Hessel

In der Regel fordern Fahrzeughersteller und Tier-1-Zulieferer von ihren Lieferanten Simulationsmodelle, um frühzeitig Systemimplementierungen durch einen virtuellen Prototypen verifizieren zu können. Solange beide das gleiche Tool verwenden, ist der Modellaustausch einfach zu realisieren. Problematisch wird es, wenn die verwendeten Tools nicht identisch sind. Der OEM musste darauf hoffen, dass das vom Zulieferer verwendete Modellformat entweder direkt oder durch automatische Konvertierung verarbeitet werden konnte. War diese Bedingung nicht erfüllt, blieb dem Auftragnehmer

nichts anderes übrig, als sich das vom OEM verwendete Tool zu beschaffen, eigene Mitarbeiter dafür auszubilden und das Modell noch einmal in der vom Kunden gewünschten Modellierungssprache zu erstellen. Forderten mehrere Kunden Modelle in verschiedenen Formaten an, konnte dies sehr zeit- und kostenintensiv werden.

Seit seiner Einführung im Jahr 1999 hat der Standard IEEE 1076.1, auch bekannt unter dem Namen VHDL-AMS (Very high speed integrated circuit Hardware Description Language – Analog and Mixed Signal), enorme Fortschritte gemacht. Der systemunabhän-

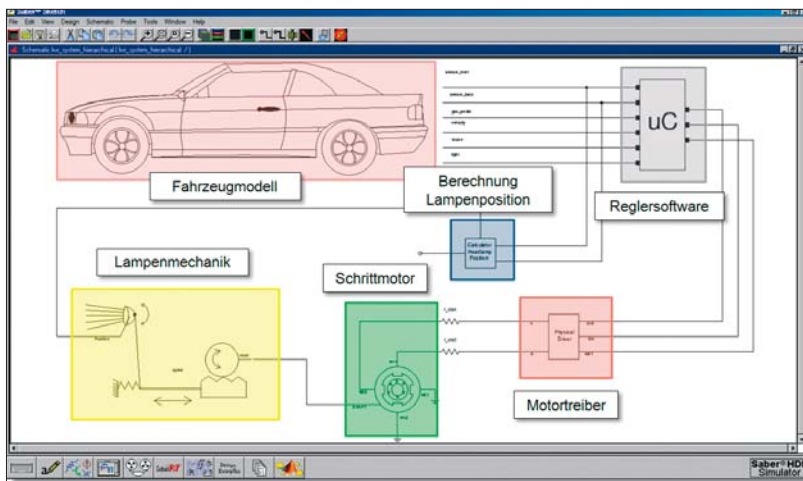
gige Sprachstandard bietet sowohl Zulieferern als auch OEMs eine gemeinsame Basis für ihre Entwicklungswerkzeuge. Er erleichtert den Austausch von Simulationsmodellen, wenn die beteiligten Geschäftspartner nicht die gleichen Simulationswerkzeuge benutzen. Voraussetzung für den reibungslosen Austausch ist, dass die beteiligten Tools den VHDL-AMS Sprachstandard absolut konform unterstützen.

### Der Sprachstandard VHDL-AMS

Der Standard IEEE 1076.1 ist eine Erweiterung des 1076-VHDL-Standards, der in Europa für den digitalen Entwurf die bevorzugte Lösung ist. Die Sprache VHDL-AMS ist eine Obermenge von VHDL, somit kann ein VHDL-AMS-Simulator auch jedes VHDL-Design verarbeiten. Aus simulationstechnischen Gesichtspunkten bietet VHDL-AMS folgende Möglichkeiten:

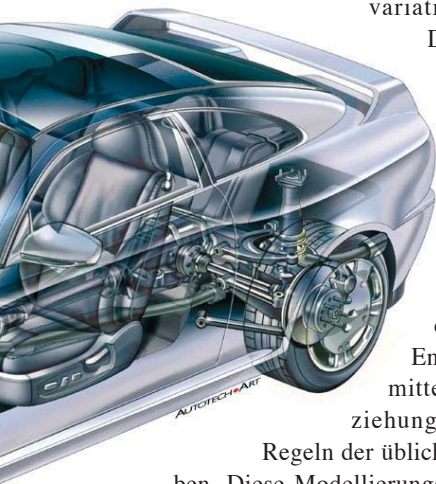
- ▶ Gleichzeitige analoge (zeitkontinuierliche) und digitale (zeitdiskrete) Simulation.
- ▶ Frequenzbereichssimulation (z.B. AC-Analyse).

Zusätzlich werden in einigen Tools erweiterte Analysen wie Parameter-



! Bild 1. Gesamtmodell einer Leuchtwertenregelung in SaberSketch, bestehend aus VHDL-AMS-Teilmodellen.

(Bildquellen: Synopsys)



variationen oder FFT unterstützt.

Die Sprache bietet das Fundament; die Bereitstellung der Analyse-Funktionen durch das Simulationswerkzeug ist weiterhin notwendig.

Eine der Stärken von VHDL-AMS sind die zur Verfügung stehenden Modellierungsmethoden. Für den klassischen Hardware-

Entwickler ist die Modellierung mittels Strom und Spannung beziehungsweise der Kirchhoffschen

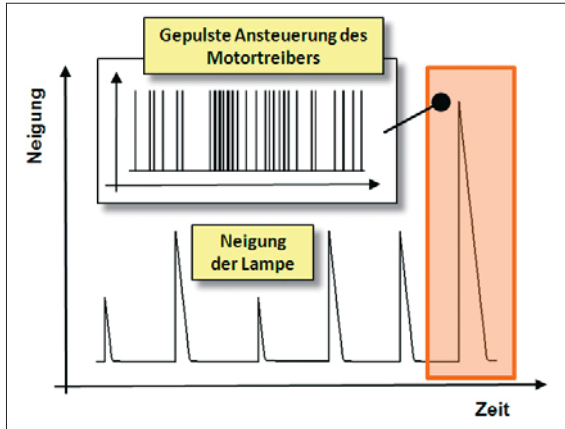
Regeln der übliche Weg, Modelle zu beschreiben. Diese Modellierungsmethode wird auch als konservativ bezeichnet, sie beruht auf dem Prinzip der Energieerhaltung.

Eine weitere Möglichkeit von VHDL-AMS ist die signalflussorientierte Modellierungsmethodik. Sie wird zur Modellierung von regelungstechnischen Systemen bzw. Algorithmen oder für die Beschreibung von kompletten Systemen auf sehr abstraktem Niveau eingesetzt. VHDL-AMS vereint die Vorteile beider Modellierungstechniken in einer Sprache mit dem Zusatz, dass sowohl zeitkontinuierlich als auch zeitdiskret simuliert werden kann. Darüber hinaus bietet VHDL-AMS die domänenübergreifende Simulation, die bei mechatronischen Applikationen notwendig ist. Diese Anwendungen vereinen mehrere Ingenieursdisziplinen in einer Applikation. Damit kann ein komplettes System entwickelt und simuliert werden.

## ■ Die VHDL-AMS-Modellbibliotheken des VDA

Der Automotive-Bereich hat die Vorteile dieser Sprache für den Systementwurf frühzeitig erkannt und unter Führung der FAT (Forschungsvereinigung Automobiltechnik) den Arbeitskreis 30 (AK30) gebildet. Dieser beschäftigt sich mit der Simulation von Automotive-Applikationen. Bei der Anwendung von VHDL-AMS hat sich schnell herausgestellt, dass die Vereinbarung einer gemeinsamen Modellierungssprache alleine nicht ausreicht, um gemeinsam an einem Projekt zusammen zu arbeiten. Deshalb wurden zunächst einige Richtlinien getroffen. Bei Anwendung dieser Richtlinien wird aus dem erstellten Modell-Code per Knopfdruck automatisch eine Dokumentation des Modells erstellt. Die Systemmodelle von Automotive-Applikationen werden meist aus kleineren diskreten Modellen zusammengestellt. Um diese Komponenten nicht immer neu zu erstellen und einen verifizierten Basisstand an Grundmodellen zu Verfügung zu haben, hat der AK30 eine tool-unabhängige Modellbibliothek in VHDL-AMS erstellt. Die Modellbibliothek enthält folgende Bestandteile:

- ▶ Spice-Bibliothek mit generischen Modellen gemäß SPICE3F5 (Spice2VHD).
- ▶ Halbleiterkomponenten nach Datenblatt (Spice2VHD\_Devices).
- ▶ Grundmodelle (Quellen, Funktionen, Signalwandler, Fundamentals).
- ▶ Typische Automotive-Modelle, wie z.B. Batterien (Automotive).



**I Bild 2. Ergebnisse der Leuchtweitensimulation.**

Eine weitere Modellbibliothek, Hybrid\_VDA, die Grundmodelle für die Simulation von Hybridfahrzeugsystemen enthält, befindet sich im Aufbau. Die gesamte Modellbibliothek wird regelmäßig auf verschiedenen Simulatoren erprobt, vom AK30 erwartet und bei Bedarf erweitert.

### ■ Standardisiertes statistisches Werkzeug

In der Automobil-Entwicklung gewinnt das Thema „Robust Design“ immer mehr an Bedeutung. Dabei sind statistische Methoden notwendig, die das Gesamtsystem gegenüber Variationen durch Einflüsse wie Temperatur, Fertigungs- und Bauteiltoleranzen absichern. Die Sprache VHDL-AMS selbst bietet hier keine direkte Unterstützung. Daher wurde die Erstellung eines „Package“ notwendig, das die benötigten Funktionen für statistische Untersuchungen bereitstellt. Die grundlegenden Arbeiten dazu fanden im AK30 statt und wurden anschließend durch die SAE abgeschlossen und im SAE-Standard J2748 umgesetzt.

Das „Package“ wird in Verbindung mit der üblicherweise in allen Simulatoren enthaltenen Monte-Carlo-Analyse für statistische Untersuchungen herangezogen. Es ist automatisch Bestandteil der Saber-Entwicklungsumgebung. Des Weiteren bietet ein „Statistical Package“ Funktionen zur Modellierung von Normal- bzw. Gleichverteilung. Sie sind das übliche Handwerkzeug für den Entwickler, um z.B. Toleranzen von Komponenten wie Widerständen oder Kapazitäten zu modellieren.

### ■ Know-how-Schutz durch Verschlüsselung

Neben dem einfachen Modellaustausch bietet der Sprachstandard eine standardisierte Verschlüsselung. Sie ist für einen reibungslosen Modellaustausch unverzichtbar, damit Hersteller ihre IP schützen und

ohne Bedenken an Kunden weitergeben können. Bei einem tool-übergreifenden Austausch wird in diesem Fall wieder ein Standard notwendig. Die Verschlüsselung ist direkter Bestandteil der Sprache. Dazu werden VHDL-AMS-Direktiven verwendet, um „Encryption Envelopes“ in das Modell einzubauen.

### ■ Simulation einer Leuchtweitenregelung

Ein typischer Anwendungsfall ist die Leuchtweitenregelung zur optimalen Ausleuchtung des Fahrbahnbereichs vor dem Fahrzeug unter wechselnden Fahrbedingungen. Das Gesamtsystem der Regelung stellt ein mechatronisches System dar, das in vereinfachter Form als Simulationsmodell in Saber (Bild 1) simuliert wird. Über das Fahrzeugmodell werden die Daten, resul-

tierend aus realen Messbedingungen, für den Fahrzyklus in das Modell eingespeist. Über zwei Fühler an den Achsen wird die Fahrzeugneigung ermittelt. Ein Algorithmus berechnet die Signale für den Treiber, der einen Schrittmotor mit elektrischer Leistung versorgt. Der Schrittmotor stellt dann über ein Umlenkgetriebe die Neigung der Lampen ein.

Um die Kosten für einen zusätzlichen Sensor einzusparen, der die aktuelle Lampenposition erfasst, wird mittels eines Algorithmus der aktuelle Ist-Wert im Stile eines „Beobachters“ rekonstruiert und an den Regler gemeldet. Somit gilt es Software, Elektronik und Mechanik in einer gemeinsamen Anwendung zu simulieren. Bild 2 zeigt die aus der Simulation resultierenden Signale. Die sägezahnähnliche Kurve zeigt die Neigung einer Fahrzeuglampe, die dem sich ständig ändernden Fahrbahnprofil angepasst wird. Die pulsierende Ansteuerung für den Treiber des Schrittmotors ist in vergrößerter Form dargestellt. Die Simulation ermöglicht eine unmittelbare Verifizierung des Algorithmus des Leuchtweitenreglers in Verbindung mit der mechatronischen Aktorik, bevor das System in Hardware existiert. Dies erlaubt einen signifikanten Zeitvorsprung in der Entwicklung, da der Algorithmen-Entwickler für die Verifikation nicht warten muss, bis ein entsprechender Prototyp des Lampensystems für Testzwecke zur Verfügung steht. ms



**Dipl.-Ing. Thorsten Gerke**

studierte Maschinenbau an der Universität Duisburg. Seit 2002 ist er bei Synopsys für die Bereiche Applikation und Presales für Automotive-Lösungen zuständig und leitet seit 2006 das Technische Marketing der Saber-Produktfamilie in Europa.

[thorsten.gerke@synopsys.com](mailto:thorsten.gerke@synopsys.com)



**Dipl.-Ing. Ewald Hessel**

studierte Nachrichtentechnik an der RWTH Aachen. Seit 1974 ist er im Geschäftsbereich Elektronik bei der Hella KGaA tätig. Dort arbeitet er an der Erprobung und Einführung neuer Entwicklungs- und Simulationsmethoden für Kfz-Steuerelektronik. Seit 2003 ist er Sprecher des AK30 in der FAT (VDA).

[ewald.hessel@hella.com](mailto:ewald.hessel@hella.com)