

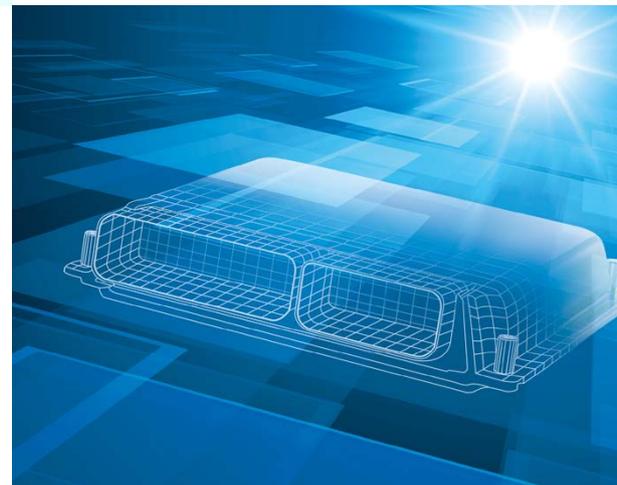
Nach vorne

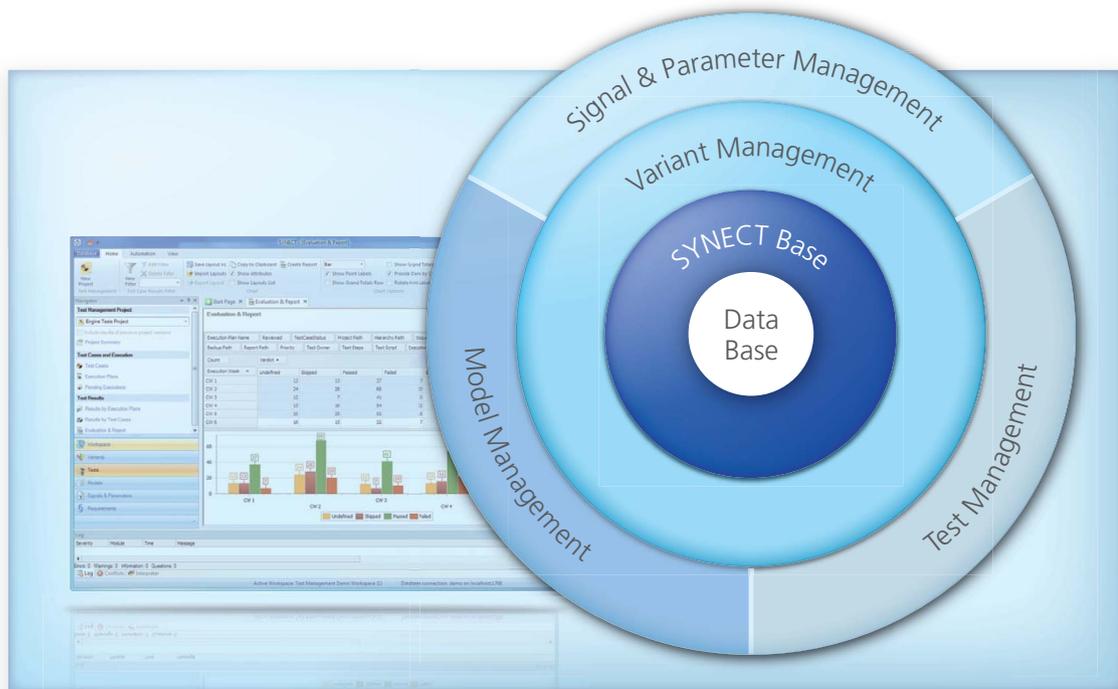
Die Produktivität der automotiven Software-Entwicklung profitiert seit vielen Jahren von der modellbasierten Entwicklung mit automatischer Seriercode-Generierung und X-in-the-Loop-Testmethoden. Entwicklung und Qualitätssicherung der rasant wachsenden Zahl neuer, software-intensiver Fahrzeugfunktionen wären ohne Werkzeugunterstützung nicht denkbar. Es stellt sich die Frage, wie sich dieser Trend in Zukunft fortsetzen wird.



Wege zu mehr Produktivität
im Entwicklungsprozess

denken





Modulares Konzept für das Datenmanagement mit dSPACE SYNECT.

Status quo

Entwicklungswerkzeuge werden heute in der Regel nicht mehr „stand-alone“ von einzelnen Entwicklern genutzt, sondern als integrale Bestandteile des Entwicklungsprozesses im Verbund mit anderen Werkzeugen, über Abteilungsgrenzen hinweg und in weltweiten Teams. Die Arbeitsabläufe und das Zusammenspiel der Werkzeuge sind in den letzten Jahren erheblich verbessert worden, nicht zuletzt auf Basis von Standards wie ASAM und AUTOSAR. Kann zukünftig eine weitere Steigerung der Produktivität erwartet werden? Und wenn ja, welche Voraussetzungen sind dafür zu erfüllen? Zunächst einmal: Höhere Produktivität kann verstanden werden als „mehr Funktionen in derselben Zeit entwickeln“. Allerdings sollte man nicht außer Acht lassen, dass bei weiter steigender Komplexität der Fahrzeugsysteme auch der heute erreichte Stand nicht von alleine aufrecht erhalten bleibt. Gerade die Entwick-

lung zukünftiger Fahrerassistenzsysteme, die den Weg zum teil- und vollautomatisierten Fahren bereiten, sowie die Elektromobilität stellen besondere Herausforderungen dar, auch mit Blick auf die zusätzlichen Anforderungen an die funktionale Sicherheit. In diesem Sinn scheint eine weitere Steigerung der Produktivität unumgänglich zu sein, selbst wenn dies „nur“ bedeutet, gleich viele Funktionen wie bisher in derselben Zeit serienreif zu entwickeln.

Konzepte zur Beherrschung der Komplexität

Fragt man Entwicklungsingenieure und Führungskräfte in der Automobilindustrie, was zu tun ist, um die Komplexität und Kosten der Software-Entwicklung im Griff zu behalten, werden meist ein oder mehrere dieser Punkte genannt:

- Mehr simulieren, d.h. mehr Details und komplexere Systeme nachbilden, dazu vorhandene Simulationsmodelle systematisch wiederverwenden.

- Steuergeräte-Softwarestände kontinuierlich erzeugen und testen, auch im Verbund.
- Früh mit der Validierung der Systeme beginnen, mehr Tests von der Straße ins Labor verlegen, dazu Aufbau einer virtuellen Absicherungsstrategie.
- Strategien für die Wiederverwendung von Modellen, Tests, Software-Komponenten und anderen Daten über Entwicklungsphasen und Teams hinweg einführen.

Um diese Empfehlungen umzusetzen, sind derzeit zwei erfolgversprechende Initiativen in den Unternehmen zu erkennen:

- Aufbau eines aktiven Managements für die schnell wachsende Menge an Modellen, Tests und anderen Datenobjekten in der modellbasierten Entwicklung. Stichworte sind Variantenmanagement, Dokumentation und Auffindbarkeit von Modellen und Tests, Bereitstellung von Anwendbarkeitskriterien (wel-



Die Komplexität sicher beherrschen durch aktives, zentrales Datenmanagement.

ches Modell, welcher Test ist wofür nutzbar), Traceability.

- Etablierung einer Prozesskette für die virtuelle Absicherung, die mit der Validierung von Funktionen und Software in einer PC-basierten Simulation beginnt und die Wiederverwendung von Modellen und Tests beim späteren Hardware-in-the-Loop (HIL)-Test von Steuergeräten erlaubt.

Datenmanagement – eine Notwendigkeit

Daten-Backbones für die modellbasierte Entwicklung und den Steuergeräte-Test sind in der IT-Infrastruktur bei Fahrzeugherstellern und Zulieferern bisher die Ausnahme. Dass Lösungsdruck besteht, erkennt man an Aussagen und Fragen Betroffener:

- „Es ist zu befürchten, dass unsere Ingenieure in der Datenflut der verschiedenen Entwicklungsphasen manchmal den Überblick verlieren.“

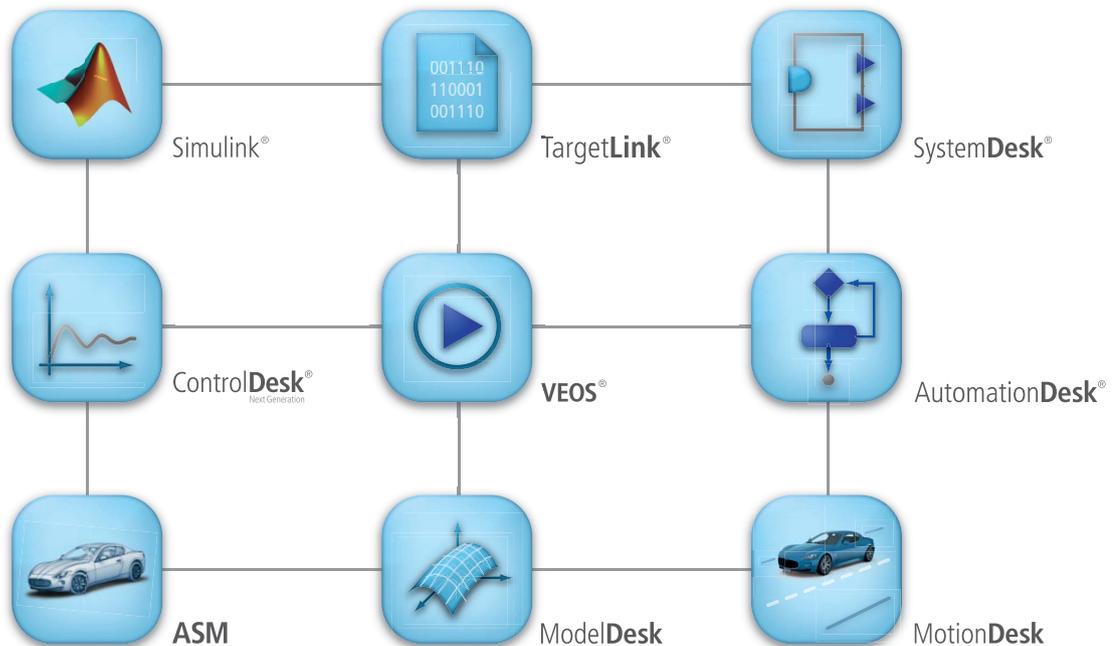
- „Wir haben immer größere Mühe, uns in der explodierenden Anzahl von Software-Varianten zurechtzufinden.“
- „Wir benötigen volle Traceability, um einen sicherheitsgerichteten Entwicklungsprozess nach ISO 26262 aufzusetzen.“
- „Mein Team verliert zu viel Zeit mit der Übertragung von Daten von einem Tool zum anderen.“
- „Wie können wir die spezifischen Datenobjekte der modellbasierten Entwicklung effizient ablegen und wiederfinden?“
- „Wie finde ich die passenden Tests für meine Steuergeräte-Varianten, um am HIL-Prüfstand keine Zeit mit nicht anwendbaren Tests zu vergeuden?“

In vielen Unternehmen wurden Aktivitäten gestartet, um eine Lösung für diese Herausforderungen zu finden. Vielerorts wurden und werden punktuelle Eigenentwicklungen betrieben, oft mit bescheidenem Erfolg. Vor al-

lem mit Blick auf die langfristige Wartung und Erweiterungen bei neuen Anforderungen stoßen Eigenentwicklungen schnell an ihre Grenzen. In den bestehenden PLM (Product Lifecycle Management)- und ALM (Application Lifecycle Management)-Werkzeugen können die „Artefakte“ der modellbasierten Entwicklung und ihre Beziehungen nicht in der benötigten Granularität abgebildet werden. Typische Datenobjekte sind Funktionsmodelle, Streckenmodelle (in verschiedener Modellierungstiefe), Source- und Object-Codes, Parametersätze, Signalbeschreibungen, Topologie-/Architekturbeschreibungen, AUTOSAR-Objekte, Testszenerien, Testskripte, Testergebnisse, Stimuli.

Anforderungsbeispiel: Datenmanagement für Modelle

Gerade am Beispiel Modelle wird deutlich, was das Datenmanagement leisten muss: Modelleigenschaften wie Schnittstellen, Parameter, Variantengültigkeit sowie anwenderspezifische



Toolkette für die virtuelle Absicherung.

sche Informationen müssen feingranular abgelegt werden, um valide Modelle für die Simulation, den Steuergeräte-Test oder für Software-Builds zusammenstellen zu können. Erst wenn ein Modell nicht nur als „Black-Box“ betrachtet wird, sondern Modularisierung und Hierarchie innerhalb des Modells zum Ausdruck kommen, lassen sich Modelle über alle Ebenen hinweg wiederverwenden, vom Integrationsmodell auf Fahrzeugebene bis zu einzelnen Blöcken einer Bibliothek. Traceability erfordert, dass ein Teilmodell oder ein Signal auf jeder beliebigen Hierarchieebene mit der Anforderung verbunden werden kann, die seiner Spezifikation zugrunde liegt. Für die Testautomatisierung möchte man ggf. die Anwendbarkeit einer Testvariante an den Wert eines be-

stimmten Modellparameters binden. Das geht weit über die dateibasierte Ablage von Modellen oder Tests in heutigen Konfigurationsmanagement-Systemen hinaus.

Datenmanagement in der modellbasierten Entwicklung

Mit SYNECT® entsteht bei dSPACE derzeit eine modulare Lösung für diese Anforderungen. Auf Basis eines integrierten Variantenmanagements werden unter anderem Test- und Modellmanagement unterstützt. Engineering Tools beispielsweise zur Erstellung und Bearbeitung von Modellen, zur automatischen Code-Generierung und zur Testentwicklung werden so angebunden, dass jederzeit definierte und konsistente Datenstände für die tägliche Arbeit

bereitstehen und kontrolliert zurückgespielt werden können. Selbstverständlich kann SYNECT in eine bestehende IT-Infrastruktur eingebunden werden. Der Austausch von Daten etwa mit ALM/PLM-Tools wird über Schnittstellen wie OSLC (Open Services for Lifecycle Collaboration) erfolgen. Bidirektionale Traceability kann so feingranular wie notwendig implementiert werden. Zum Beispiel lassen sich Beziehungen zwischen Anforderungen und daraus abgeleiteten Funktionsmodellen, Tests und Testergebnissen vollständig nach ISO 26262 dokumentieren. Das Ergebnis ist eine deutliche Verbesserung der Datenkonsistenz, effizientere Zusammenarbeit und einfachere Wiederverwendung der Daten im gesamten Entwicklungsprozess.



Absicherungsstrategien mit virtuellen Steuergeräten

Um einen höheren Absicherungsgrad in der virtuellen Welt zu erreichen, ist eine frühe und kontinuierliche Software-Integration der beteiligten Steuergeräte anzustreben, so dass diese einzeln oder im Verbund als „virtuelle Steuergeräte“ mit aussagekräftigen Umgebungsmodellen in einer PC-Simulation getestet werden können. Ingenieure können auf diese Weise sehr früh die Performance einer komplexen, über mehrere Steuergeräte verteilten Funktion erleben. Fehler in der Regelstrategie oder in der Software-Implementierung werden früh gefunden (Kosten-

tional Mock-up Interface)-Standard integriert. Besonders effizient und mächtig wird die PC-Simulation dadurch, dass in Verbindung mit VEOS sämtliche Test- und Experimentierwerkzeuge eingesetzt werden können, die auch am HIL-Prüfstand zur Verfügung stehen. Umfangreiche Simulationsläufe und Tests werden am PC in der gewohnten Tool-Umgebung entwickelt, in Betrieb genommen und ausgeführt. Modelle werden bereits hier konfiguriert, parametrisiert und validiert. Sämtliche Modelle, Tests und Daten können dann mit denselben Tools am HIL-Prüfstand wiederverwendet werden, so dass dieser von „nicht-produkti-

Die virtuelle Absicherung schafft Zeit- und Kostenvorteile.

und Zeitersparnis). Da die Simulation nicht unter Echtzeitbedingungen laufen muss, können detailliertere Umgebungsmodelle und aufwendigere Simulationsverfahren zum Einsatz kommen als später am HIL-Prüfstand, um Funktionen so realitätsnah wie möglich zu optimieren.

Die Simulationsplattform VEOS

Für den Einstieg in die virtuelle Absicherung bietet dSPACE das Produkt VEOS® an, eine PC-Simulationsplattform für virtuelle Steuergeräte, verteilte Funktionen und Umgebungsmodelle. Virtuelle Steuergeräte werden typischerweise aus Software-Komponenten gemäß AUTOSAR-Standard generiert, können aber auch direkt aus Simulink®/TargetLink®-Funktionsmodellen erstellt werden. Basis-Software-Module wie Services, Betriebssystem und Kommunikationsstacks können zur realistischen Darstellung des Steuergeräte-Verhaltens hinzugenommen werden. Umgebungsmodelle aus verschiedenen Modellierungstools werden über den neuen FMI (Func-

ven“ Aufgaben wie der Testentwicklung entlastet wird.

Nahtloser Übergang zum realen Steuergeräte-Test

Virtuelle Steuergeräte können am HIL-Prüfstand als Ersatz für fehlende echte Steuergeräte oder sogar als „Device under Test“ mitlaufen. Damit lassen sich in einem Steuergeräte-Verbund nach und nach alle virtuellen durch reale Steuergeräte ersetzen, so dass der Übergang zum Test des realen Steuergeräte-Netzwerks möglichst reibungslos vonstattengeht. Eine solche einheitliche Toolkette für den Test virtueller und realer Steuergeräte erlaubt es Fahrzeugherstellern und Steuergeräte-Zulieferern, neue Absicherungsstrategien zu definieren und auszurollen. Das Potential, das diese Erweiterung der Testmethodik für die automotiv Software-Entwicklung mit sich bringt, wurde in ersten Projekten mit Fahrzeugherstellern nachgewiesen. ■

Dr. Rainer Otterbach,
dSPACE

Fazit

Um die Produktivität in der automotiv Software-Entwicklung bei der enormen Zunahme hochwertiger, softwarebasierter Fahrzeugfunktionen auch in Zukunft beizubehalten oder noch weiter zu steigern, müssen neue Wege zur Beherrschung der Komplexität beschritten werden. Aktives Datenmanagement und virtuelle Absicherungsstrategien sind zwei zielführende Ansätze, die auf den bisher erreichten Stand in der modellbasierten Entwicklung aufsetzen und ein hohes Potential zur weiteren Optimierung der Entwicklungsprozesse bieten. dSPACE freut sich darauf, auch in Zukunft innovative Wege gemeinsam mit seinen Kunden zu gehen und auf Basis der neuen Produkte SYNECT und VEOS bedarfsgerechte Lösungen einzuführen.

Dr. Rainer Otterbach

Dr. Rainer Otterbach ist Leiter Produktmanagement bei der dSPACE GmbH.

